Coordinating Human-Robot Teams with Dynamic and Stochastic Task Proficiencies

RUISEN LIU*, MANISHA NATARAJAN*, and MATTHEW C. GOMBOLAY, Georgia Institute of Technology

As robots become ubiquitous in the workforce, it is essential that human-robot collaboration be both intuitive and adaptive. A robot's ability to coordinate team activities improves based on its ability to infer and reason about the dynamic (i.e., the "learning curve") and stochastic task performance of its human counterparts. We introduce a novel resource coordination algorithm that enables robots to schedule team activities by 1) actively characterizing the task performance of their human teammates and 2) ensuring the schedule is robust to temporal constraints given this characterization. We first validate our modeling assumptions via user study. From this user study, we create a data-driven prior distribution over human task performance for our virtual and physical evaluations of human-robot teaming. Second, we show that our methods are scalable and produce high-quality schedules. Third, we conduct a between-subjects experiment (n=90) to assess the effects on a human-robot team of a robot scheduler actively exploring the humans' task proficiency. Our results indicate that human-robot working alliance ($p < 0.001$) and human performance ($p = 0.00359$) are maximized when the robot dedicates more time to exploring the capabilities of human teammates.

CCS Concepts: • **Computing methodologies → Planning under uncertainty**; **Multi-agent planning**; • **Human-centered computing → Empirical studies in HCI**; *HCI theory, concepts and models*.

Additional Key Words and Phrases: Scheduling; Human-Robot Teaming; Optimization

ACM Reference Format:
Ruisen Liu, Manisha Natarajan, and Matthew C. Gombolay. 2020. Coordinating Human-Robot Teams with Dynamic and Stochastic Task Proficiencies. *Proc. ACM Meas. Anal. Comput. Syst.* 37, 4, Article 111 (August 2020), 37 pages. https://doi.org/TBD

1 INTRODUCTION

Advancements in robotics have opened opportunities for humans and robots to collaborate within joint workspaces, particularly in final assembly manufacturing. Effective collaboration requires coordination and utilizing the distinct abilities of each team member. A human-robot team should be able to leverage its unique capabilities to achieve safe, effective, and fluent coordination. As the Internet of Things (IoT) and collaborative robots become ubiquitous, we have the first opportunity to capture and explicitly reason about this uncertainty to develop high-quality, ad hoc teaming mechanisms in which robots can learn the relative strengths of their human counterparts to efficiently seek and reason about the best team composition and task assignment.

Robots may excel at repeating certain tasks, while humans typically have latent, dynamic, and task-specific (i.e., a "learning curve") proficiencies. However, enabling a robot to infer human strengths and weaknesses while

*Both authors contributed equally to this research.

Authors' address: Ruisen Liu, ruisenericliu@gatech.edu; Manisha Natarajan, mnatarajan30@gatech.edu; Matthew C. Gombolay, matthew.gombolay@cc.gatech.edu, Georgia Institute of Technology, 801 Atlantic Dr, Atlanta, Georgia, 30318.

2476-1249/2020/8-ART111 $15.00
https://doi.org/TBD

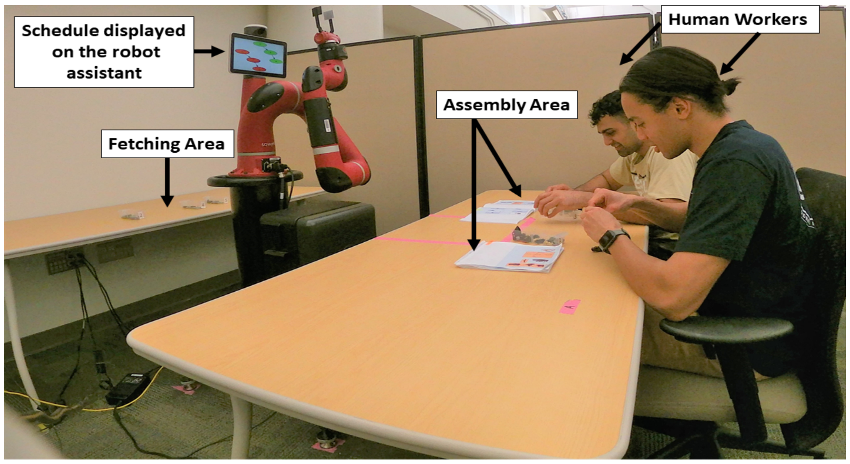


Fig. 1. This figure depicts the workspace of our human subject experiment.

ensuring that the team satisfies requisite scheduling constraints is challenging. First, humans naturally exhibit variability in task performance, driven by variance or error in task execution behavior. This variability introduces uncertainty in estimating relative task performance while planning to meet individual tasks and joint task deadlines. Second, human task performance also exhibits time-extended variability due to learning effects with practice, resulting in uncertainty on future performance in a finite horizon. Although we may expect an individual to improve at a task over time, estimating the short and long-term horizon in comparative efficiency by committing a particular agent to repeat a task is challenging [59]. Third, a lack of consideration for human preferences and perceived equality may, in the long run, put efficient behavior and fluent coordination at a contradiction [29, 32].

While a human-centered approach may be able to determine ideal task assignments while balancing efficiency and preference, it is also a tedious, manual process. For example, a new assembly line may not reach peak performance until six months of ramping up during which human workers are honing their skills and adjusting team roles. Another example of a setting where an automated, predictive approach would be helpful is in seasonal manufacturing, where temporary workers must be quickly yet optimally assigned in a narrow window of opportunity to fulfill task demands. Effectively, an automated, predictive approach requires 1) means to estimate the efficiency and robustness of any possible assignment for human agent-task combinations and 2) mechanism to trade-off the value of trying out new task assignments to actively learn more about the latent characteristics of different human workers versus maintaining the status quo (i.e., exploiting the estimation of future human proficiency with current assignments). We desire a scalable algorithmic approach that can factor in the stochasticity of human behavior for fluent human-robot teaming.

Recent advances in scheduling methods for human-robot teams have shown a significant improvement in the ability to dynamically coordinate large-scale teams in final assembly manufacturing [31, 33, 54]. Prior approaches typically rely on an assumption of deterministic or set-bounded uncertainty for each worker-task assignment in perpetuity. However, by reasoning myopically, these methods fall short in explicitly reasoning about the uncertainty dynamics in human work that could result in significant productivity gains. Given these limiting modeling assumptions, these methods are unable to give guarantees – probabilistic or otherwise – of satisfying the upper- and lower-bound temporal constraints in the presence of this uncertainty. Without such probabilistic guarantees, we cannot bound the uncertainty of a schedule and guarantee a consistent range of resulting behavior.

In this paper, we formulate a novel, human-aware, computational method for scheduling human-robot teams that synthesizes robust schedules while adapting to the characteristic behavior of participants as they learn skills while following a schedule. Our approach considers the distinct problem of dynamic uncertainty attributed to the human ability to improve at tasks through repeated practice. In particular, we develop a framework in which a robot reasons about the strengths of its teammates by trying out novel, possibly sub-optimal allocations of work, while still providing a probabilistic guarantee on satisfying the temporal requirements of the manufacturing process.

To accomplish successful human-robot interation (HRI) and collaboration with improved performance and team alliance, we develop computational methods to determine high-quality schedules and adaptions to human performance, with contributions that 1) infer dynamic characteristics about individual human performance and their ability to learn on the task, leading to improved efficiency in task allocation, and 2) enable an efficient computation method for evaluating the robustness of candidate schedules with respect to deadlines, which enables 3) a fast optimization algorithm for iterative improvement of human-robot schedules based on observed progress. To the best of our knowledge, our approach is the first to reason explicitly about risk allocation with respect to deadlines while also tackling dynamic uncertainty and projecting human learning capability.

In addition, we show our approach to task scheduling also yields qualitative improvements in human-robot teaming. Prior work has shown that switching tasks among team members can lead to greater efficiency and generalizability to new task variants [53, 58]. Further, the effectiveness of human-robot collaboration relies heavily on building trust and team fluency among teammates [17, 31, 68]. However, these works have not considered the significantly more challenging problem of robust coordination of human-robot teams under temporal upper- and lower-bound constraints with time-varying stochastic task proficiency. Our work seeks to improve human-robot teaming while optimizing for efficiency during scheduling with probabilistic guarantees of meeting temporal deadlines. To demonstrate simultaneous improvements in both areas, we design and conduct an experiment to evaluate trade-offs in exploration-versus-exploitation for maximizing overall team performance while also inferring how trust and human-robot team alliance varies as a function of the degree of exploration.

Our paper provides the following contributions:

1. We create a data set of human task proficiency to build stochastic models of learning rates and task durations ($p < 0.05$) characterizing human task performance.
2. We leverage these models to enable a fast computation method for a tight upper-bound for schedule duration.
3. We derive an adaptive update mechanism to predict each individual's future task performance.
4. We show these individual components enable evolutionary search as a viable method for real-time schedule exploration and optimization in simulation. Our approach results in aggregate makespan improvements of $44.8 \pm 6.4\%$ and $18.9 \pm 5.5\%$ for 20 and 50 interdependent tasks respectively, relative to our baseline.
5. We conduct a human-subject experiment ($n = 90$) in an analogue manufacturing setting that yielded statistically significant improvements for human-robot team alliance ($p = 0.01$) and in human performance with regards to task completion times ($p = 0.0035$) when consistently factoring task diversity over more myopic approaches.

2 BACKGROUND

In this section, we discuss existing work at the intersection of human-robot teaming and schedule optimization. First, we cover desirable behavioral characteristics of human-robot teaming, and how such criterion have been explored in task allocation and teaming thus far. Then we examine approaches to schedule optimization under uncertainty, and the limitation of existing model formulations in translating such approaches to human-robot scheduling. We summarize the intersection of our work across human-robot team coordination facets compared to others in Table 1.

	Team Fluency	Physical HRI Experiment	Human Performance Estimation	Temporal Constraints	Scalable Real-Time Scheduling
Our Approach	✓	✓	✓	✓	✓
Claure et al., 2019	✓		✓		
Chen et al., 2018	✓	✓			
Gombolay et al., 2018		✓		✓	✓
Rahman & Wang, 2018	✓	✓			
Tsarouchi et al., 2017	✓	✓			
Huang et al., 2015	✓	✓			
Nunes & Gini, 2015				✓	✓

Table 1. This table provides a summary of facets found in recent work in human-robot team coordination.

2.1 Characteristics of Human-Robot Teaming

For collaborative human-robot teams, scheduling requires reasoning about explicit and implicit preferences by human teammates [31, 61, 74]. In addition, team fluency and trust are important elements to evaluate and maximize during collaboration of human-robot teams [15, 39, 40]. Our objective is to consider a robot agent as the primary source of decision making in order to formulate a schedule. In addition to the heuristic driven challenges posed in Section 1, Gombolay et al. [31] found in a study of human-robot teams that while people inherently value human teammates more than robotic ones, providing robots with authority over team coordination resulted in objective improvements in efficiency and also perceived team fluency. The improvement in efficiency likely resulted due to increased tendencies for humans to assign a disproportionate amount of work to themselves rather than allocating to a robot. In addition, such designation of authority resulted in increased perceived value of robotic teammates, and positive correlations in team fluency and a human subject's willingness to collaborate with robots. Direct delegation can help provide team cohesiveness, as an intermediate robot liaison may hinder perceived human team member inclusion [65].

Groom and Naas [37] list traits for successful human-robot teammates which include sharing a common objective, comprehending and fulfilling each other's roles, and trust. Further, they provide examples of several user studies in human-robot interaction to establish people's trust in robots as a crucial factor in determining the success of the human-robot team. Trust between humans is different from human-robot trust as humans perceive automated systems and robots to be more credible and are hence much more sensitive to errors in such systems [48]. Prior studies have shown that user's trust in robots is dependent on several factors such as task performance [60], user experience [80], and rate of failure [23, 63].

Previous literature also suggests that an integrated and proactive approach to human-robot interaction would be received well by human teammates. Shah et al. [68] designed *Chaski*, a robot plan execution system that chooses and schedules the robot's actions to adapt to the human partner, and mininizes the partner's idle time. Chao and Thomaz [15] designed a system based on timed Petri Nets for multimodal turn-taking and joint action meshing. Their corresponding human subject study found that participants who assigned to the "interruption" condition rated their interaction as more natural when working in human robot teams. In a study on human-robot collaboration in a joint workspace via handovers, Huang et al. [40] designated proactive and reactive coordination strategies to enable a robot to adapt to a human receiver's requirements. It was found that a proactive approach to adaptation lead to the greatest levels of team performance and user experience.

2.2 From Task Allocation to Scheduling on Human-Robot Teams

We define the primary distinction between task allocation and scheduling as the inclusion of inter-agent temporal, precedence, and synchronization constraints [45]. A key contribution of our work in schedule optimization technique is missing a built-in process to reason about changing human capabilities, and thus constraints, over time. Effective teaming requires robots to interact and generate human-aware behaviors while planning [78]. Some frameworks formulate objectives for multiple additional criteria, such as fatigue, quality, cycle time, and fairness, but do not explicitly target optimization for the duration of a schedule [18, 19]. Kwon and Suh [46] abstract the challenge of schedule optimization to one of maintaining proactive robot behavior in the face of uncertainty. Gombolay et al. [32] found that humans would prefer to work with robots that factor in their preferences, but not necessarily so if at the expense of team. In addition, they found that providing robots authority on team coordination strongly improved the perceived value of these agents.

Claure et al. [19] shows that fairness is a significant factor for ideal task allocation, fairness, where fairness was defined as constraining the minimum rate of human teammate being selected. Claure et al. also factor in the relative difference of human skill level in the distribution of resources for virtual computer-based tasks. However, this work does not consider time-extended scheduling with upper- and lower-bound temporal constraints and inter-task dependencies. Further, this work does not seek to explicitly infer a model of the probability distribution over humans' task completion times [55].

Prior work for coordinating human-robot teams primarily focuses on discerning behavior that maximizes ideal behavioral traits for human-robot teaming without factoring in the challenges of robust schedule optimization, especially under temporal upper- and lower-bound constraints with heterogeneous time-varying human task proficiency. Our work looks to generate fluent human-robot teaming while simultaneously optimizing for efficiency during scheduling with probabilistic guarantees for meeting scheduling deadlines. As such, we propose modifications to chance-constrained scheduling approaches that can handle risk distribution for a large number of tasks and leverage information provided by the team of agents, and proactively adjust the schedule to fit expectations.

2.3 Robust Multi-agent Schedule Optimization

Traditional scheduling methods apply a deterministic approach, with numerical values representing the estimated duration of tasks [33, 54]. These operate under the assumption that the optimization problem is aiming at joint coordination of multiple machines or vehicles that are brought together [55]. Classic examples of multi-agent task allocation include truck dispatching, job shop scheduling, and transit routing [2, 21, 49].

Simple Temporal Networks (STNs) are commonly used as models for optimization of multi-agent problems [3], 15]. These models are popular because their constraints can be validated in polynomial time [20, 56, 79]. STNs admit at most one interval constraint on any pair of time intervals, with a default assumption of deterministic estimate on task duration. By nature, human beings are not deterministic and such approaches fail to adequately characterize human preferences and behavior. In many real-world applications, such an assumption is also incompatible with a notion of uncertainty with respect to the duration of each task [13]. Extensions on STNs have been used to include uncertainty sets for multi-agent problems [8, 9, 41]. However, few extensions actually model uncertainty as probabilities [28, 73]. In addition, although these methods reason about set-bounded uncertainty, they do not reason about the model or people behind the uncertainty set [22]. Instead, they define a hard upper bound for the output.

Alternatively, there exists decentralized methods for scheduling, where methods for task allocation with temporal constraints via auctioning [24, 51]. However auctioning only considers the iterative sale of tasks and can be considered a reactive approach [38, 72] reasoning about uncertainty. Typically, these approaches allocate one task at a time to a team of agents [44, 47]. As a result, these methods are not well suited for proactive reasoning about uncertainty through parametric estimation of agent capabilities.

Other approaches seek to reason about a parametric model for the uncertainty set. These proactive approaches seek to reason about the robustness of uncertainty sets, which define the set of possible values generated by a model [35, 62]. Robust optimization techniques look to satisfy all constraints based on some measure of robustness [35]. A probabilistic approach is commonly accepted in methods for robust analysis [4, 12, 43, 70] and synthesis [75–77]. Robust synthesis seeks to find a feasible or optimal solution by introducing multipliers or scaling variables to enable formulation relaxation [6, 26, 64].

However, these type of relaxations present a computational challenge, as the compute time of a mixed-integer linear program (MILP) solver grows exponentially with respect to the number of tasks and agents [34]. As the number of potential agent-task combinations grows to large-scale real-world settings, such approaches become computationally intractable. It is often found that synthesis require a large number of samples in order to achieve the desired probabilistic guarantee [75, 76]. Although model parameterization may help alleviate sampling costs, there exists an inherent trade-off between sampling requirements and conservative relaxation. Linear optimization models for the uncertainty set lead to too conservative bounds, while non-linear models led to too expensive computation costs [5, 25, 69].

Chance-constrained programming, also known as probabilistic constrained programming, is a separate optimization approach that guarantees the probability of meeting a set of constraints to be greater than $1 - \epsilon$ [16]. A logical step for scheduling problems with multiple, sequential constraints is to consider the joint satisfaction of constraints by distribution of the risk, ϵ, for the overall scheduling problem amongst the various constraints [11, 50, 52]. Yet application of existing chance-constrained approaches towards scheduling distributes risk-satisfaction over $O(n)$ individual tasks, resulting in high conservatism as well.

Soyster et al. sought to define a linear optimization model to construct a feasible solution to convex data optimization problems but resulted in too conservative of an approach [69]. Improvements on this approach [69] used ellipsoid models for the uncertainty set, which minimized conservatism but led to nonlinear and non-convex models with expensive computation [5, 25]. Subsequently, Γ-Robust optimization was proposed to preserve Soyster's linear framework for cheaper computation, but achieved equivalent bounds, specifically for sums of symmetrically distributed random variables [7]. In Γ-Robust optimization, at most, Γ entries will deviate from their expected value [7].

Evolutionary-based search approaches were also considered for generating estimates for robustness, within the context of process scheduling for individual machines. One approach estimated the robustness of the schedule based on the efficiency of viable schedules in the local neighborhood of the initial candidate schedule [42]. Other approaches sought to find solutions that were robust to processing disruptions and variable processing times [14, 67]. These approaches offer alternative routes to evaluating robustness without computing the uncertainty set and avoid the trade-off problem between sample-requirements and excessive conservatism. However the consequence of not explicitly evaluating the uncertainty set is an inability to provide a theoretic guarantee to robustness.

Our work draws across disciplines to revisit the sample cost and conservatism trade-off while providing means for fluent exploration and quick discovery of latent proficiency for human task execution in tightly temporally constrained schedules. We introduce a new upper bound mechanism for schedule makespans that limits conservatism without high sample requirements to develop an *a priori* model for expected task duration. We then adapt methods from control theory to provide quick and precise updates to estimates of individual task proficiency, within the flow of repeated schedule optimization. Finally, we show how these methods can be combined within a evolutionary-based search method to enable efficient schedule optimization while improving human-robot team fluency in a user study.

3 PROBLEM FORMULATION

We formulate our objective function to minimize the latest finish time of all tasks in the schedule (i.e. the makespan) while penalizing a schedule based on the number of repetitions for agent-to-task assignment combinations (entropy) as a means to actively learn and explore latent human task proficiency. We explicitly reason about each human and robot's individual strengths and weaknesses in task completion and ensure that natural variability in human performance is accounted for to provide a probabilistic guarantee for meeting all task deadlines. Our objective function is subject to time-based constraints related to probabilistic task completion times, waits, upper- (deadlines) and lower- (i.e. wait constraints) bound temporal constraints.

Let us consider schedule optimization after n_s rounds of schedule implementation. Let n_t denote the number of tasks and n_a denote the number of agents. Let τ_i denote the i_{th} task. Let $x_{i,j}$ denote the relative order of tasks. Let τ_i^s, τ_i^f denote the start and finish times of each task. Let A_i^a be a binary variable indicating the assignment of agent $a \in A$ iteration of task τ_i. Let $r_i^{a,n}$ be the number of repetitions that agent a has completed for task τ_i in the past n_s rounds. Let $PDF_{i,a,n}$ denote the probability distribution function for the duration of a task τ_i conditional of the number of task repetitions (n) the assigned agent (a) has performed thus far. Additional let us denote $PDF_{\tau_i^f}$ as the probability distribution function for the finish time of task τ_i when following the schedule. We define the corresponding cumulative distribution functions as $CDF_{i,a,n}$ and $CDF_{\tau_i^f}$ accordingly.

$$\min z = z_1 + \lambda z_2 \tag{1}$$

$$z_1 = \max_{\tau_i \in \tau} \tau_i^f \tag{2}$$

$$z_2 = \frac{1}{n_t n_a} \sum_{\tau_i} \left[\sum_a | \left(\frac{1}{n_a} \sum_a r_i^{a,n} \right) - r_i^{a,n} | \right] \tag{3}$$

$$A_i^a \in \{0,1\}, \forall i \in I, \forall a \in A \tag{4}$$

$$\sum_{a \in A} A_i^a = 1, \forall i \in I \tag{5}$$

$$\tau_i^f - \tau_i^s \geq 0, \forall i \in I \tag{6}$$

$$\tau_i^f, \tau_i^s \geq 0, \forall i \in I \tag{7}$$

$$\left(\tau_j^s - \tau_i^f\right) x_{i,j} A_i^a A_j^a \geq 0, \forall i, j \in I, \forall a \in A \tag{8}$$

$$x_{i,j} + x_{j,i} = 1, \forall i, j \in I \tag{9}$$

$$\tau_j^s - \tau_i^f \geq W_{i,j}, \forall i, j \in I | W_{i,j} \in TC \tag{10}$$

$$ub_{\tau_i} \geq \tau_i^f - \tau_i^s \geq CDF_{i,a,n}^{-1}(1 - \epsilon_{\tau_i}), \forall i \in I, \forall a \in A, \sum_{\tau_i \in \tau_i} \epsilon_{\tau_i} = \epsilon \tag{11}$$

$$D_{i,j}^{rel} - CDF_{\tau_j^f - \tau_i^s}^{-1}(1 - \epsilon_D) \geq 0, \forall \tau_i, \tau_j \in \tau | \ \exists D_{i,j}^{rel} \in TC, \sum_{D \in TC} \epsilon_D = \epsilon \tag{12}$$

$$D_i^{abs} - CDF_{\tau_i^f}^{-1}(1 - \epsilon_D) \geq 0, \forall \tau_i, \in \tau | \ \exists D_i^{abs} \in TC \wedge \sum_{D \in TC} \epsilon_D = \epsilon \tag{13}$$

We define our objective function in Equation 1 as a linear combination of minimizing the makespan in Equation 2 and entropy in Equation 3. Equation 4 states that an agent is either assigned or unassigned to a task. Equation 5 requires all tasks τ_i to be assigned to an agent. Equations 6 and 7 ensure that each task, τ_i, has a non-negative duration and that the start, τ_i^s, and finish, τ_i^f, times are likewise non-negative. Equations 8 and 9 require each agent to complete one task at a time. Equation 10 requires a minimum wait time of $W_{i,j}$ between the end of task τ_i and the start of task τ_j.

To optimize subject-to-chance constraints for our uncertainty in task completion time, we need to distribution risk by tasks (Equation 11), or by deadlines (Equations 12-13). If we enforced task-based risk allocation, we define a lower bound by guaranteeing that the task succeeds at a rate of $1 - \epsilon_{\tau_i}$. The upper bound would then be defined by the earliest deadline associated with that task. However, we elect enforce a deadline-based risk allocation rather than a task-based risk allocation to reduce the risk allocation spread of ϵ from $O(n)$ tasks to $O(D)$ deadlines. As such, we enforce allocation with Equations 12 and 13. Equation 12 requires the likelihood of completing relative deadline $D_{i,j}^{rel}$ between the start of task τ_{i_n} and the end of task τ_j be greater than $1 - \epsilon_D$. Equation 13 requires that the likelihood of completing each absolute deadline D_i^{abs} be greater than $1 - \epsilon_D$.

3.1 Technical Challenges in Exact Optimization

There are two primary challenges that prohibit a MILP solver from directly addressing the overall problem using this formulation.

(1) **This formulation does not provide any adaptive reasoning about individual agent capabilities to complete tasks.** It is necessary for the programmer to define the probability distribution function for each agent-iteration-task duration combination $PDF_{i,a,n}$ each time. Thus within this framework, we have no inherent mechanism that takes in agent samples and updates a belief on $PDF_{i,a,n}$. Consequently, maintaining the same schedule misses opportunities to optimize on inter-agent skill capabilities.

(2) **The probability distribution function for task finish time $PDF_{\tau_i^f}$ and risk allocation ϵ_D in in Equations 12 and 13 are under-defined.** It is not possible to compute $PDF_{\tau_i^f}$ until a full assignment has been made for all tasks and task orders, $\{A_i^a, x_{i,j}\}$, associated with requisite tasks leading up to the task deadline. Knowing the full set of task assignments is also required to compute the probability distribution of the makespan, the computational multiplier in any search algorithm. We elaborate on the computational expense for each probability distribution function $PDF_{\tau_i^f}$ in Section 6 and show that it cannot be expressed in non-integral form. In addition, it is up to the programmer to define the risk distribution for ϵ_D. We discuss our alternative solution for upper-bounding $PDF_{\tau_i^f}$ for real-time evaluation.

Below, we detail our methods to tackle these challenges and enable new search methodologies for our dynamic scheduling process.

4 COORDINATION ALGORITHM OVERVIEW

To generate an adaptive method that improves upon optimization with fixed estimates for human behavior, we seek an approach that evaluates dynamic, improving human task completion times and merges them with fixed robot task completion times. Without being able to directly use a MILP solver given the challenges noted in Section 3.1, we require alternative solution mechanisms that will construct and evaluate a large number of candidate schedules. Given the computational expense to evaluate a single schedule makespan, we must derive and leverage a sufficiently fast upper-bound evaluation method. In addition, we need a reliable parametric update to the expected agent-specific task duration as we observe agents over time (i.e., a learning curve model with updates for each distinct agent as they perform tasks).

In the following sections, we first validate two parametric assumptions in Section 5 that are used to parameterize human task completion times. Then, in Section 6, we offer a fast upper-bound calculator as a substitute for the objective value to achieve a fast estimation of the robustness and makespan of any candidate schedule. Subsequently, we reason about agent capabilities by computing a parametric update to the estimate for a person's likely future task completion time in Section 7. Finally, we utilize these two methods to facilitate an iterative search process for finding a schedule that balances minimizing the time needed for completion and exploring new task-agent assignment pairs, while staying robust to deadlines. We depict our search process in Figure 2.

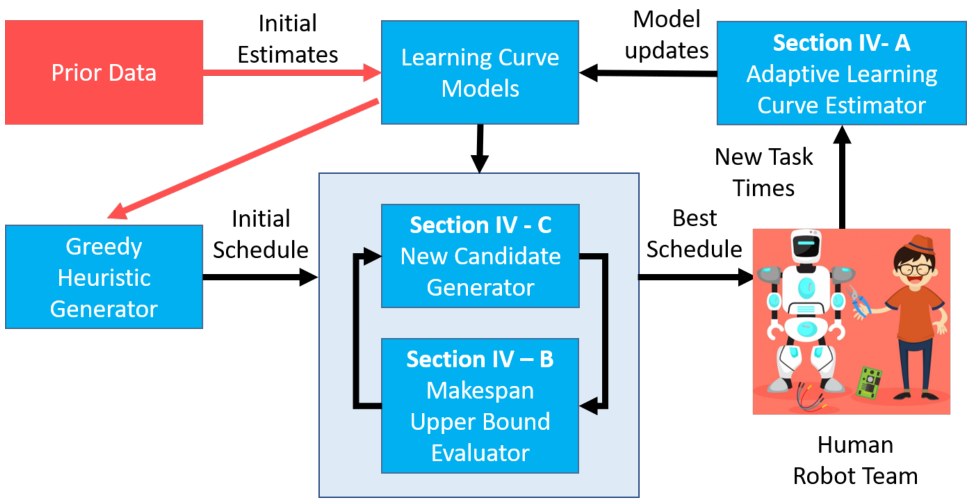

Prior Data
Initial Estimates
Learning Curve Models
Model updates
Section IV- A Adaptive Learning Curve Estimator
New Task Times
Greedy Heuristic Generator
Initial Schedule
Section IV - C New Candidate Generator
Section IV – B Makespan Upper Bound Evaluator
Best Schedule
Human Robot Team

Fig. 2. A flowchart depicting our iterative process for candidate generation and task proficiency comprehension during schedule optimization for human-robot teams.

Algorithm 1: Optimization workflow pseudocode

```
agentParameters ← compute(populationPriors) ;
bestSchedules ← greedyHeuristic(agentParameters) ;
for N iterations of scheduling do
    while time ≤ timeLimit do
        candidateSchedules ← searchMethod(bestSchedules) ;
        bestSchedules ← upperBoundEvaluator(candidateSchedules) ;
    end
    bestSchedule ← bestSchedules[0] ;
    newTaskTimes ← apply(bestSchedule) ;
    agentParameters ← update(newTaskTimes) ;
end
```

For each round of scheduling, we generate initial candidate schedules by applying a greedy heuristic designed for deterministic task scheduling. Then, we apply one of two distinct any-time methods we develop (Section 8) for improving the schedule. This evaluation process is aided by the fast upper bound evaluation detailed in Section 6. Once a new optimal schedule is selected, we pass the schedule to the human-robot team for completion. The duration of each executed task is recorded and used to update the parameters of each corresponding assigned agent's learning curve, as detailed in Section 7. The learning curve is then used to re-project estimations for the expected duration of future tasks. Our approach is also depicted in Algorithm 1.

5 ALGORITHM ASSUMPTIONS AND VALIDATION

We hypothesize two assumptions about human task completion capabilities which we validate through a user study.

H1: *Humans learn over time and that their learning curve follows an exponential function of generic form $y = c + ke^{-\beta i}$ over the course of multiple iterations, where i is the iteration and c, k, β are parameters [59].*

H2: *For a population of people completing an assembly task, the population distribution of task times approximates a normal distribution.*

To validate the two assumptions, we conducted a human-subject pilot study to collect data for learning curves by recording the time taken by different subjects to physically assemble LEGO kits of varying difficulty. In the pilot study, each participant was assigned to complete six different tasks for five iterations. We recruited 18 participants ranged from age 22 to 31 (Median:24, Male:11, Female:7) to participate in the pilot study. Figure 3 shows the task completion times from the pilot study.

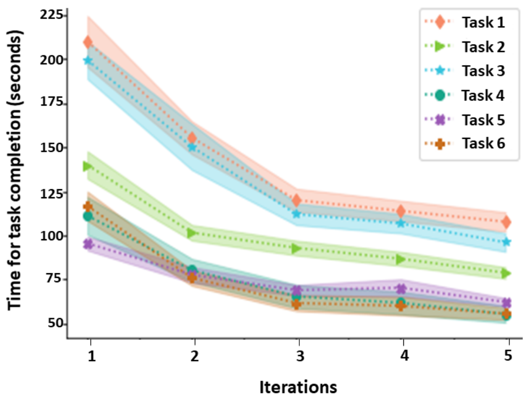


Fig. 3. Task completion times from pilot study

Task	c	k	β
1	101.83 (4.29)	239.94 (29.16)	0.78 (0.09)
2	80.12 (2.86)	148.81 (21.76)	0.91 (0.12)
3	87.30 (4.45)	218.55 (20.8)	0.66 (0.11)
4	53.94 (5.34)	112.07 (17.74)	0.725 (0.15)
5	63.20 (2.69)	71.92 (16.98)	0.82 (0.25)
6	57.02 (3.30)	196.73 (36.02)	1.19 (0.14)

Table 2. c, k, β parameters from the pilot study, with standard error in parentheses.

Task	Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5
1	0.007	0.035	0.399	0.285	0.807
2	0.031	0.427	0.773	0.015	0.261
3	0.903	0.009	0.844	0.563	0.798
4	0.133	0.12	0.055	0.015	0.133
5	0.292	0.004	0.007	0.003	0.062
6	0.591	0.943	0.024	< 0.001	0.201

Table 3. p-values from Shapiro-Wilk's test for Normality Assumption

We test the normality assumption for task completion times (**H2**) on all the six tasks for five iterations using Shapiro-Wilk's test with Bonferroni-adjusted α level of 0.001 (0.05/30 as we are testing for 30 combinations). Results from Shapiro-Wilk's test find that we do not reject the null hypothesis (that the distribution is normally distributed) at the $\alpha = 0.05$ confidence level for 29/30 task-iteration combinations, indicating that 97% of the data comply with our normality assumption. Furthermore, the average time taken per task decreased exponentially (**H1**) as a function of the number of iterations, as suggested by literature [59]. We report the c, k, β values used to fit the pilot data distribution in Table 2. As such, we are confident in our two modeling assumptions in generating simulated schedules and tasks.

6 FAST UPPER BOUND FOR SCHEDULE ROBUSTNESS

In Section 3.1, we stated that one primary challenge of the MILP formulation was under-defined chance constraints in Equations 12 and 13 and for evaluating the makespan in Equation 2. In this section, we elaborate of this computational difficulty and develop an analytical upper-bound to address it.

First, we define the schedule as robust if it is possible to complete all tasks by the desired deadlines with probability $(1 - \epsilon)$. An approach to distributing risk across deadline success (S_i) can be obtained via Boole's inequality, as shown in Equation 14, which enables expression of the joint chance constraints of n deadlines as shown in Equation 15.

$$Pr[\bigcup_{i=1}^{n} S_i] \leq \sum_{i=1}^{n} Pr[S_i], \tag{14}$$

$$Pr(S_i) \geq (1 - \epsilon_i), \sum_{i=1}^{n} \epsilon_i \leq \epsilon \tag{15}$$

Boole's inequality allows us to select the appropriate ϵ_D for each deadline in Equations 12 and 13 given an overall schedule risk allocation ϵ. To compute the robustness given that ϵ_D, we need to evaluate the probability distribution for the completion of the last task associated with that deadline.

We then define the distribution for task completion time $PDF_{\tau_i^f}$. We can express $PDF_{\tau_i^f}$ as a function of the time elapsed before the start of the task due to preconditions and the duration of the task as shown in Equation 16.

$$PDF_{\tau_i^f} = max\left(\{PDF_{\tau_i^f}^{precon} + w_i^{precon}\}, PDF_{\tau_{i-1}^f}\right) + PDF_{i,a,n} \tag{16}$$

In Equation 16, $PDF_{\tau_{i-1}^f}$ is the end time of the previous task completed by the current agent, and $\{PDF_{\tau_i^f}^{precon} + w_i^{precon}\}$ denotes the set of distributions that correspond with satisfaction of each task precondition and wait constraint. This equation requires two computations: a maximization (prior precondition tasks in parallel) and an addition (new task in series) as show in Figure 4.

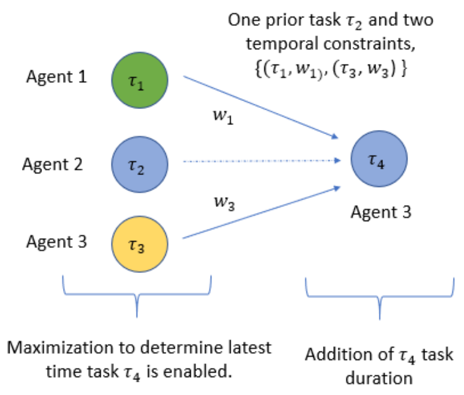

One prior task τ_2 and two temporal constraints, $\{(\tau_1, w_1), (\tau_3, w_3)\}$
Agent 1
Agent 2
Agent 3
τ_1
τ_2
τ_3
τ_4
w_1
w_3
Agent 3
Maximization to determine latest time task τ_4 is enabled.
Addition of τ_4 task duration

Fig. 4. This figure depicts a visualization of operations resulting from temporal constraints. Agent 2 is completing tasks 2 and 4, but is dependant on Agents 1 and 3 to complete task 4's preconditions.

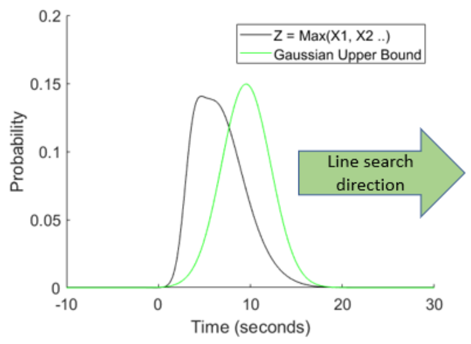


Fig. 5. This figure depicts a Gaussian-based upper-bound on the underlying distribution of a random variable Z such that $Z = \max(X_1, X_2, X_3)$.

Generally, the maximization of a set of random variables $Y = max\{X_i\}_{i=1,2..n}$ yields a resulting probability distribution $Y \sim f_y(y)$ which can be expressed by Equation 17.

$$f_y(y) = \sum_{i=1}^{n} f_{x_i}(y) \prod_{j=1, j\neq i}^{n} F_{x_j}(y) \tag{17}$$

The subsequent probability distribution resulting from the sum of two random variables $Z = Y + X$, $Z \sim f_z(z)$ can be expressed by Equation 18.

$$f_z(z) = \int_{-\infty}^{\infty} f_x(x) f_y(z - x) dx \tag{18}$$

Recall that task finish times PDF_{ian} are normally distributed. The sum of two normally distributed random variables has a quick closed-form solution. If $X \sim N(\mu_x, \sigma_x^2)$ and $Y \sim N(\mu_y, \sigma_y^2)$, then Z can be parameterized by $Z \sim N(\mu_x + \mu_y, \sigma_x^2 + \sigma_y^2)$. However, the maximum of normally distributed random variables given by Equation 17 does not have a non-integral solution, unless $\exists y \in \mathbb{R}\ s.t.\ F_{x_i}(y) \approx 0, F_{x_j}(y) \approx 1\ for\ i \in N, j \neq i$. That is, that the rightmost normal distribution does not "overlap" with the distributions.

For any task in the schedule that has preconditions, the resulting expression for the probability distribution function of the task finish time $PDF_{\tau_i^f}$ can only be evaluated via integration. To find the distribution associated with the deadline, it is necessary to traverse the directed acyclic graph (DAG) of the schedule and its associated task sequence to ascertain the expected completion time of each task along the way. Consecutive multiplication and integration computations quickly become expensive, and this expense is compounded by the need for re-calculation with each new schedule assignment $\{A_i^a, x_{i,j}\}$ while searching for an optimal schedule.

As the computation cost scales exponentially in time with the number of tasks and schedules to evaluate, we instead search for a fast approximating Gaussian distributed upper bound for Equation 17 and introduce a small amount of conservatism in return for a scalable evaluation. We empirically demonstrate the efficacy of our trade-off in Section 9.2. We apply line search for a Gaussian distribution that upper bounds $max\left(\{PDF^{precon}_{\tau^f_i} + w^{precon}_i\}, PDF_{\tau^f_{i-1}}\right)$ in Equation 16. Formally, for $Y = max\{X_i\}_{i=1,2..n}$, we search for a distribution $f_g(y)$, such that Equation 19 holds.

$$F_g(y) \leq F_y(y) \; \forall y \in [0, \infty] \tag{19}$$

We now assert and prove Lemma 1 that this distribution exists.

Lemma 1. *There exists a Gaussian cumulative probability distribution $g \sim N(\mu, \sigma)$ s.t. Equation 19 is true with probability $1 - \epsilon$, for arbitrarily small ϵ.*

Proof. For $F_y(y), \exists\, b \; s.t. \; (1 - \epsilon) \leq F_y(b)$. For an arbitrary $\sigma, \exists\, N(\mu, \sigma), \; s.t. \; \mu - k\sigma \geq b, F_N(\mu - k\sigma) \leq \epsilon$, for $\mu, k \in [0, \infty]$. Since ϵ is arbitrarily small, then $F_n(\mu - k\sigma) \leq \epsilon < 1 - \epsilon \leq F_y(b) \leq F_y(\mu - k\sigma)$. □

Through experimentation, we found a good initial guess for upper-bound distribution is $f(g)(y) \sim \mathbb{N}(\mu = max(\mu_{x_i}), \sigma = mean(\sigma_{x_i}))$, which is equivalent to selecting the mean of rightmost distribution. With the distribution average, we can then perform line search for incremental shift of μ and σ by step sizes α and β, respectively, until Equation 19 is satisfied as shown in Figure 5. If α is scaled with respect to σ/c, the maximum steps needed is a single-digit multiplier of c. Checking the exit condition at each step can be inexpensive, since although $f_y(y)$ has no closed-form solutions, $F_{x_i}(y)$ in $F_y(y) = \prod_i^n F_{x_i}(y)$ can be evaluated with a single look-up. Given the Gaussian distribution's symmetric properties, the number of evaluation points to produce a tight bound can be empirically as few as 12 points spaced uniformly within $[\mu - 3\sigma, \mu + 3\sigma]$.

In summary, by computing a maximization and then a summation operation for each task, we can obtain the probability distribution function over each task's finish time $PDF_{\tau^f_{i-1}}$ via Equation 16. However, the resulting distribution is expensive to represent and compute. Thus, we efficiently search for a Gaussian upper-bound for $PDF_{\tau^f_{i-1}}$ that introduces a small amount of conservatism in exchange for propagating efficient upper bound computation for all tasks in the schedule. Subsequently, we can check if the distribution of the final task before each deadline satisfies the risk ϵ_D allocated via Equation 15. Finally, we can compute the makespan distribution by evaluating the probability distributions' maximum associated with each agent's final task. We analytically show the advantageous trade-off of our approach in Section 9.2.

7 LEARNING CURVE UPDATE ALGORITHM

With risk allocation and an effective upper-bound for $PDF_{\tau^f_{i-1}}$ for deadline constraints defined, we address the other challenge of modeling humans' learning capability on a particular task by parametrically reasoning about a human's unknown learning curve. Given our validation of **H1** and **H2** in Section 5, we can expect to model the learning curve as an exponential function of generic form $y = c + ke^{-\beta i}$, where c, k, β are parameters and i is the repetition number. However, humans exhibit additional variance in task completion time. It is difficult to tease out how much a reduction in task completion time is due to chance from steady-state variance versus an actual improvement in performance associated with a learning effect. While we can compute a model prior by looking at population averages, we need a way to tailor a model update for each individual.

We design our learning curve model update as an adaptive Kalman filter [1], where our state vector is composed of learning curve parameters and our observation consists of observed task duration. We detail our implementation in Algorithm 2. This formulation is analogous to a control model where the system dynamics are stationary and the noise model is hidden. We model each individual's learning curve parameters as a hidden but static state, with the best initial guess as the population average. After each task is completed, we obtain a new observation and update our parameters. Our primary challenge comes from determining analogous initial estimates for the state, prior covariance and covariance noise matrices, P, Q, R, and the corresponding update rate α. To find a suitable estimate, we bootstrap by repeatedly sub-sampling the population prior to computing the initial covariance matrix P and scale Q and R in proportion to P. To determine the ideal update rate for Q and R, we run repeated simulations to sweep through range $[0, 1]$ in intervals of 0.25 and find $\alpha = 0.9$ to be on average the best update rate.

Algorithm 2: Agent learning curve model update

```
Initialize from population $x_{k-1|k-1}$;
Initialize noise estimates $Q, R$ ;
**while** *While not done* **do**
    $x_{k|k-1} = x_{k-1|k-1}$ ;
    $P_{k|k-1} = P_{k-1|k-1} + Q$ ;
    $d_k = z_k - h_k(x_{k|k-1})$    # observation error ;
    $s_k = R + H_k^{[1]} P_{k|k-1} H_k^{[1]T}$ ;
    $K_k = P_{k|k-1} H_k^{[1]T} [s_k]^{-1}$    # kalman gain ;
    $x_{k|k} = x_{k|k-1} + K_k d_k$    # model update ;
    $\epsilon = z_k - h_k(x_{k|k})$    # residual error ;
    # adaptive noise updates ;
    $R = \alpha R + (1 - \alpha)(\epsilon\epsilon^T + H_k^{[1]} P_{k|k-1} H_k^{[1]T})$ ;
    $Q = \alpha Q + (1 - \alpha)(K_k d_k d_k^T K_k^T$ ;
    $P_{k|k} = I - K_k H_k^{[1]} P_{k|k-1}$ ;
**end**
```

8 SEARCH METHODS: MODIFIED BRANCH AND BOUND OR EVOLUTIONARY OPTIMIZATION

In this section, we discuss two search methods that could be utilized to optimize our schedule. We first discuss how a lower-bound could be formulated to pair with our upper-bound method from Section 6 to enable branch-and-bound search for our MILP formulation. In addition, we discuss how to generate genetic diversity and mutation to enable a evolutionary optimization approach to search.

8.1 Modified Branch and Bound (B&B) Optimization

MILPs are often solved via a branch-and-bound solver, which require non-trivial upper and lower-bounds for pruning sub-trees, and a branching order for binary variables. We discussed an upper bound mechanism in Section 6, and introduce in this section a non-trivial lower-bound at each node for dynamic variables in Equations 12 and 13 and a branching order for variables $\{ A_i^a, x_{i,j} \}$. We provide pseudocode for our branch and bound with Algorithm 3 and 4.

Algorithm 3: Modified branch and bound

```
UB ← computeUB(greedySchedule) ;
solution ← greedySchedule ;
taskOrder ← greedySort(tasks) ;
varOrder ← [(Aij, xij) for task in taskOrder] ;
queue ← [ Node(assignment = Null) ] ;
while timer ≤ TimeLimit AND queue !empty do
    newNode ← queue.pop() ;
    if newNode == leafNode then
        val ← computeUB(newNode.assignment) ;
        if val < UB then
            solution ← newNode.assignment ;
            UB ← val ;
    else
        childrenList ← getChildren(newNode, varOrder) ;
        for assignment in childrenList do
            LB = relaxLP(assignment) ;
            if LB < UB then
                queue.append(Node(assignment)) ;
            end
        end
    end
end
```

Algorithm 4: relaxLP(assignment)

```
Initialize taskMeans, taskStds ;
Initialize LP ;
LP.assignKnownValues(assignment) ;
LP.addNonDeadlineConstraints ;
minStd ← min( taskStds ) ;
for task, deadline in taskDeadlines do
    len ← computeLongestPath() ;
    $\sigma \leftarrow \sqrt{len * min\_std}$ ;
    LP.addDeadlineConstraint($\sigma$) ;
end
LP.relax() ;
RETURN LP.optimize() ;
```

To compute a correct lower-bound for the schedule at a node, we consider the LP relaxation of the MILP with two optimistic conditions: 1) each task $\tau_i \sim N(\mu, \sigma)$ is completed by a fast agent, and 2) there is a fast agent readily available for the task when preconditions are satisfied. We formulate condition 1) by setting the lower bound for each task in Equation 11 as $\mu - z * \sigma$, for z satisfying some significance value. For condition 2, we derive the relaxation on Equations 12 and 13 when an agent is always readily available.

Since we have an upper bound for task finish times, we can update Equations 12 and 13 to be inclusive of the start and finish times with respect to a standard normal Gaussian Φ.

$$D_{i,j}^{rel} - \left((\tau_j^f - \tau_i^s) + \left[\Phi^{-1}(1 - \epsilon_D)\sigma_{D_i} \right] \right) \geq 0, \forall i, j \in I | \exists D_{i,j}^{rel} \in TC, \sum_{D \in TC} \epsilon_D = \epsilon \tag{20}$$

$$D_i^{abs} - \left(\tau_j^f + \left[\Phi^{-1}(1 - \epsilon_D)\sigma_{D_i} \right] \right) \geq 0, \forall i \in I | \exists D_i^{abs} \in TC \wedge \sum_{D \in TC} \epsilon_D = \epsilon \tag{21}$$

To enforce a lower bound on the makespan, we wish to find a lower bound on the corresponding distribution variance $\sigma_{D_{\tau_i}}$ when agents are readily available. That is, we look to define a lower bound for $\sigma_{D_{\tau_i}}$ such that these deadlines can be met easier. Let σ_{min} be the minimum of σ_{τ_i} for all known agent task combinations tasks. Then the minimum variance at the end of task deadline, D_{τ_i}, can be found by finding the maximum set of preconditions, of length k, that need to be sequentially satisfied to reach task τ_i, Since an agent is available and task variance is σ_{min}, we see that $\sigma_{D_{\tau_i}}$ can be computed from k consecutive iterations of convolutions, as defined by Equation 18, which yields $\sigma_{D_{\tau_i}} = \sigma_{min}\sqrt{k}$.

For variable assignment we sort a task order list based on our initial schedule policy. We sort the branch order of variables by that task order and place agent-task assignment variables A_i^a before task-order variables $x_{i,j}$. If the variable is an agent-task assignment A_i^a, we prioritize agent assignments from least-to-most existing workload. If the variable is about relative task order $x_{i,j}$, we prioritize assigning True over False. Combined, these two priorities start our search with a schedule that evenly distributes tasks and the task orderings closely resemble a schedule generated by applying a greedy heuristic designed for deterministic scheduling.

8.2 Evolutionary Optimization

We note that finding a non-trivial lower bound that will accelerate the efficiency of a branch-and-bound approach by frequently pruning sub-trees. Pruning occurs by exceeding the incumbent upper-bound, which is difficult given a wide domain on uncertainty sets. Alternatively, we can search via evoluationary optimization. One of the prerequisites for this optimization is the seeding and maintenance of a diverse population. However, such a seeding process is challenging, as for a scheduling problem with inter-task dependencies (e.g. upper- and lower-bound temporal constraints), as a randomly generated schedule is highly likely to be infeasible [34]. As such, the majority, if not all, of seeded solutions are pruned from the population without contributing meaningful candidates.

To overcome this inefficiency induced by random seeding of schedules, we propose to generate initial schedules by employing a construct based on a modified "soft" EDF (earliest deadline first) schedule policy. EDF takes the average expected task duration, and utilizes Floyd-Warshall to find the corresponding sequence of tasks in order of latest start times to meet all deadlines [10]. In "soft" EDF, our objective is to provide population diversity by exchanging tasks in the sequence that may likely be started at the same time. We sort the latest start times for each task, and examine tasks that have start times $|\tau_i^s - \tau_j^s| < \delta$ for some δ much less than the average task duration. If this criterion is true for two adjacent tasks in the sort, a coin-flip is used to determine whether the order of the tasks should flipped. This generates candidate schedules with a much higher likelihood of being feasible to the original temporal constraints.

We employ an evolutionary-based approach to sequentially construct new candidates, keeping a percentile of best candidate schedules as defined by our objective function. We search via iterative mutation by swapping task allocations, task order, and crossover between agent assignment lists. In addition, we re-seed the population with diversity by employing our soft EDF based heuristic. Candidate schedules are also evaluated for robustness to deadlines; those that are not robust are eliminated. As discussed in Section 6, the computation cost for evaluating the probability distribution of a schedule makespan scales exponentially with the number of tasks and temporal constraints. Our method for defining an upper-bound empirically enables at least a 10^3 speed up in computation time, as we later show in Section 9.2. This speedup enables 10^3 more iterations of any type of search for optimizing the schedule. After successive rounds of evolution, the best schedule is selected for implementation. The exact parameters selected for our genetic algorithm are detailed in Section 9.5.

9 COMPUTATIONAL EVALUATION OF OUR COORDINATION ALGORITHMS

In this section, we simulate complex scheduling problems to confirm the general ability of our individual algorithm components to provide a fast, tight upper-bound and reason about individual capabilities.

9.1 Simulation Generation

We utilize our pilot study data from Section 5 to generate synthetic scheduling problems of high complexity and inter-agent teaming. We first define a scheduling problem with temporal constraints. We set the overall deadline to $\frac{\mu+3\sigma}{n_a}$, where μ, σ are the total expected time of all tasks and its corresponding variance and n_a is the number of agents. That is, we compose a problem in which the majority of new agents complete the schedule just in time. In addition, we add intermediate deadlines for 1/5th of all tasks in the schedule. Finally, we intertwine tasks by giving non-starter tasks a weighted probability [50%, 30%, 15%, 5%] for having [0, 3] precedence constraints. We choose to generate scheduling problems with fewer than 75 tasks and up to three agents. We based our design on prior literature [34] indicating that solutions are not tractable past the trade-off curve of those milestones.

To simulate heterogeneous agent-task capabilities, we turn to our pilot study to construct a population of agent learning curves for each task in the schedule. We generate the parameters c, k, β for each agent-task combination by comprising each parameter of three sub-distributions, which respectively represent the task, agent, and joint task-agent contributions to that parameter. We repeat to generate a proportionally scaled hidden variance at every iteration, representing the agent's potential to deviate from their learning curve. We set our parameters such that each agent-task learning curve resembles the following characteristics from our pilot study: 1) 95% of simulated agents are within 30% of the population average for the initial task duration, and 2) all agents are able to the learn the task quickly, converging to task proficiency (less than 2% improvement in further repetitions) within ten iterations.

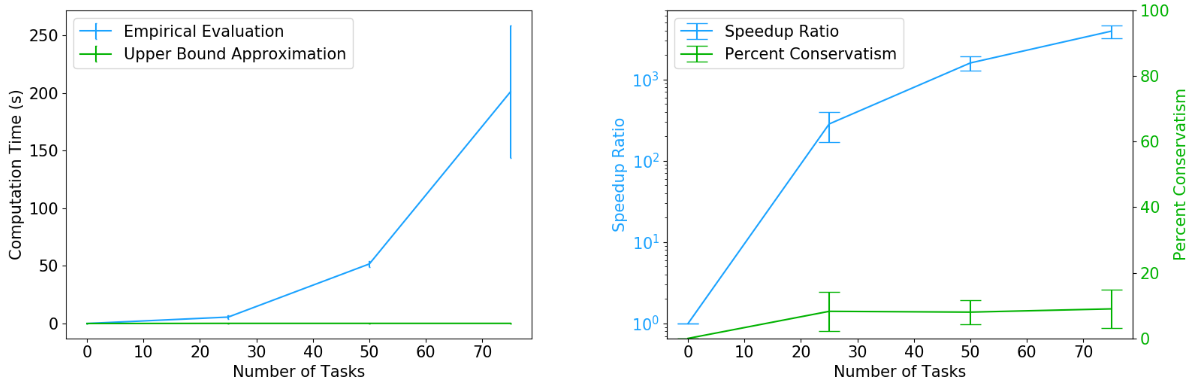
9.2 Computation Time for Robustness Evaluation

Using our simulation generation method, we constructed 30 sample scheduling problems each for three different number of tasks, i.e. $\{25, 50, 75\}$. We then generated feasible schedules for three workers to complete each task. We then evaluated the average computation time costs for deriving an accurate probability distribution for the makespan of the schedule using both numerical estimation and Gaussian upper-bound approximation given n tasks. While computing a Gaussian approximation always took less than 0.1s for 25, 50, and 75 tasks, the corresponding numerical approximation (i.e. the baseline method) took 5.56 ± 1.53s , 51.69 ± 2.53s, and 201.14 ± 57.75, as shown in Figure 6a. We show the speedup ratio between empirical evaluation and our fast upper-bound in Figure 6b. From the figure, we can see that the computational expensive for evaluating the exact probability distribution of the makespan becomes over 10^3 longer than utilizing our fast upper-upper bound approximation. For a schedule robustness requirement of 95% probability of success, the average amount of conservatism (percent added time) introduced by using our approximation method was 8.44 ± 10.1%. As such, we see that our approach for computing an upper-bound scales well for large number of tasks without introducing increasing conservatism in our bound for the makespan. Thus we can be confident that our bounding approach is critical in enabling real-time search for finding new schedules.

9.3 Efficacy of Learning Curve Update

To isolate and demonstrate our algorithm's effectiveness in predicting an individual human's learning curve, we examined the ability of our adaptive update to predict the task duration in subsequent repetitions of the same task. We look to show critical improvement over utilizing the population average as a baseline deterministic estimate. We generated 100 instances where an initial learning curve was estimated from a population of 50 humans for 20 iterations. The population was generated using the same method as our overall scheduling simulation described in Section 9.1. A new agent was then introduced with unknown parameters. For each new agent, we ran the simulation 20 times to calculate the average total error of each model in predicting the agent task duration across 20 iterations.

Across 100 instances, the baseline estimation model resulted in a median total error of 167.8s, with the first and third quartile at 87.1 and 252.6s, respectively, as shown in Figure 7. The adaptive estimation model had a median



(a) Corresponding raw compute times

(b) Speed-up ratio trade-off with conservatism

Fig. 6. These figures depict the trade-offs in employing numerical quadrature for a probability distribution over satisfying temporal constraints vs. our upperbound and the corresponding compute time.

total error of 49.4s, with the first and third quartile at 43.5s and 61.1, respectively. Crucially, our update model removed many outliers found in the population data.

On average, switching from the baseline estimate to the learned individual model decreased the total error by $59.2 \pm 25.3\%$. In addition, we see reductions in the maximum and minimum error across repetitions of each simulated instance. The average improvement in maximum error was $54.6 \pm 26.4\%$. The average improvement in minimum error was $63.0 \pm 26.4\%$. We also see that extreme outliers were also eliminated when switching from the population model to our adaptive model. As such, we are confident that our adaptive update consistently improves over the baseline estimate and enriches our prediction capacity across iterative repetitions between candidate agents.

9.4 Any-time Search Method Selection: B&B v.s. Evolutionary-Based Search

We consider the relative strength of our branch and bound and evolutionary-based search methodologies for finding better schedules. To evaluate in a non-biased manner, we compare the consecutive decrease in the makespan upper-bound as a function of search time, without parametric updates to agent-specific task-capabilities. That is, we assume a homogeneous pool of agents for one iteration of scheduling, and ascertain the ability of each search algorithm to find better schedules starting from a baseline of an EDF based schedule policy. We run 25 random scheduling problems and allow a 1-hour search on a single core.

For 20-50 task schedules, we see the evolutionary optimization algorithm consistently improve over time, while the branch and bound finds new incumbents with far lower frequency as the number of tasks increases, as shown in Figure 8. For higher number of tasks, we expect the branch and bound not to find significant incumbents in a short window of time. Since evolutionary-based search outperformed branch and bound search, we select this latter approach as the search method for both our simulated and physical human-robot interaction ($n = 90$) evaluations in Section 10.

9.5 Full Schedule Optimization

We now seek to show the combined results from assembly of our coordination algorithm components. We considered two scenarios with 20 and 50 tasks respectively where three agents are working to complete the extended set of interdependent tasks. We choose these values to highlight different levels of task inter-dependency, as per our

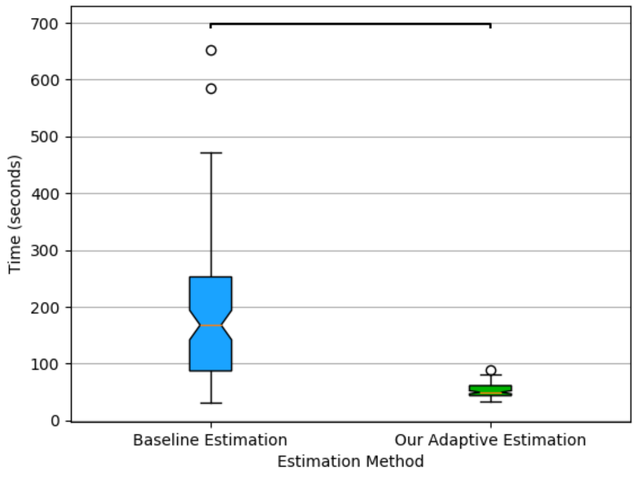


Fig. 7. This figure depicts the distribution of cumulative estimation error after 20 iterations. The baseline estimation utilizes the static population average, whereas our adaptive estimation iteratively projects new estimates on the fly. Significant difference between baseline and adaptive estimation $p < 0.01$.

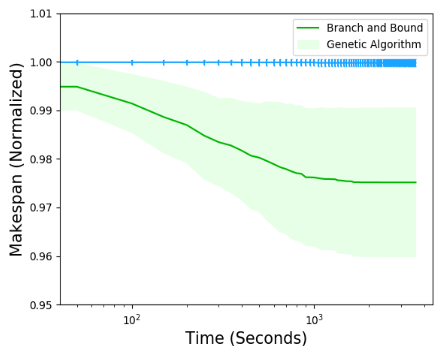
generation method described in Section 9.1, any task may have a temporal dependency with another task in the full set of tasks. For each number of tasks, we simulated 50 random scheduling problems. To demonstrate results where the three agents are heterogeneous, we draw each agent's task specific learning curve randomly from either the top or bottom quartile of the population for each task.

We recall that in all scenarios, 1) 95% of agents have untrained capabilities within 30% of the population average for all tasks, and 2) all agents are capable to task proficiency (less than 2% improvement in further repetitions) within ten iterations. This rapid improvement requires fast discovery of agent characteristics and task exchange while incurring explorations costs due to lack of exploitation of commit learning ability by the previous agent.

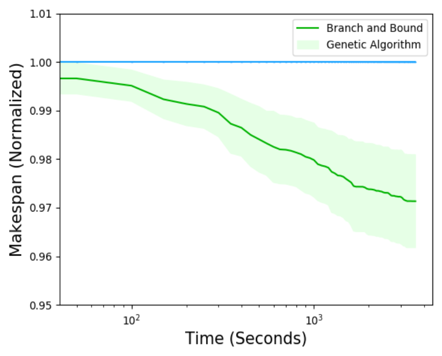
For our evolutionary-based algorithm, we initiate with a population of 90 candidate schedules via our soft EDF diversity policy mentioned in Subsection 8.2. At each iteration we allow the search to evolve for 50 generations, creating 10 new candidate samples based on our random schedule generation heuristic and 10 more via mutation. We evaluate all the candidate schedules and remove the 20 weakest schedules to maintain the population at 90 during each evolution. For weighting exploration v.s. exploitation, we linearly anneal λ over the first half of the scheduling iterations. Our starting λ coefficient is equivalent to 10% of the initial makespan. We evaluated random and EDF as heuristics for generating an initial schedule. In each of our following figures, we show the normalized makespan improvement for four different conditions:

(1) **Ours**: Our full algorithm with an exploration factor and adaptive updates.
(2) **Ours \ λ**: Our algorithm without any exploration factor.
(3) **Ours \ (λ and adaptation)**: Evolutionary optimization without any adaptive predictions.
(4) **Baseline**: Committing to the initial EDF generated schedule.

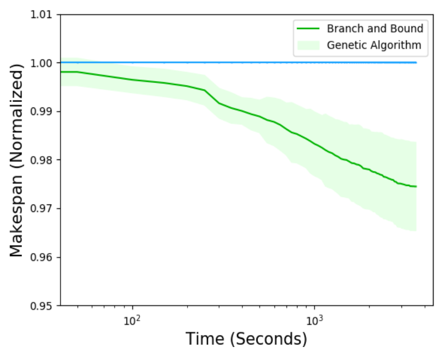
We distinguish the individual contributions of the evolutionary-based optimization, the adaptive learning curve, and the exploration factor λ by showing makespan improvement across conditions 1, 2, and 3. In addition, we are not



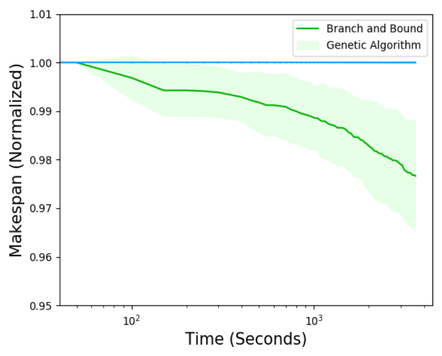
(a) 20 tasks



(b) 30 tasks



(c) 40 tasks

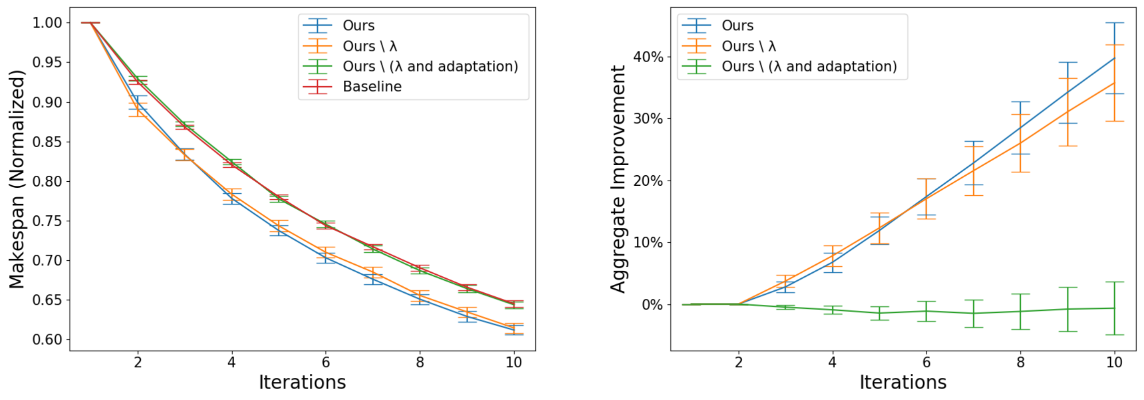


(d) 50 tasks

Fig. 8. These figures depict comparative upper bound improvements via search over 1-hour for 3 human agents and 20, 30, 40, and 50 tasks, corresponding to Figures 8a, 8b, 8c, and 8d, respectively. We normalize the results with respect to the initial starting schedule candidate.

including results in which the initial schedule is randomly generated, as there was no instance in which initiating search from a population of randomly generated schedules yielded a feasible schedule.

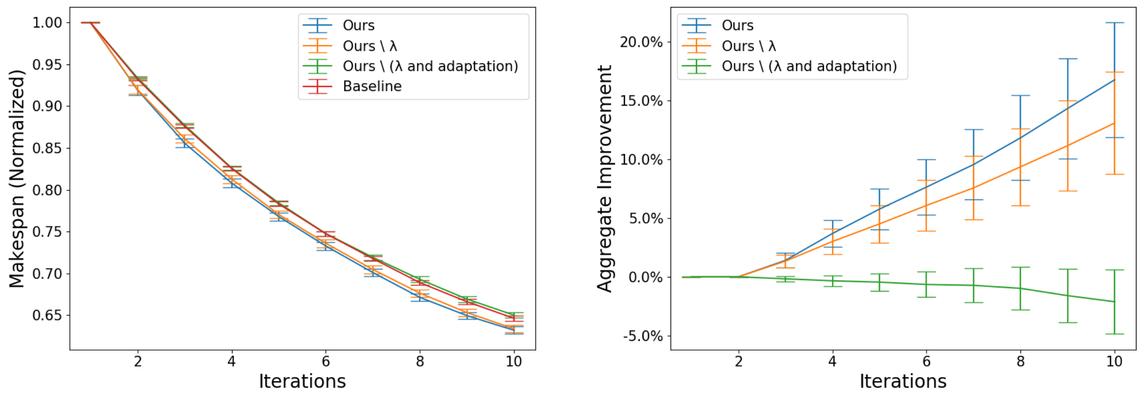
In every condition, we start with no distinct information on any agent, and thus each condition begins with a homogeneous estimation for agent capabilities. To normalize results across different scheduling problems with differing makespans, we divide all makespans by the initial round one makespan associated with the schedule generated by EDF. To illustrate the true comparative advantage on the scheduling makespan by each conditions, we empirically sample the task-duration of the final schedule selected by each algorithm. However, at run time, the algorithm only optimizes knowing a fast upper-bound derived by our method described in Section 6.



(a) Change in makespan over time

(b) Aggregate improvement over baseline

Fig. 9. These figures depict benchmark results for 20 agents and 3 tasks, makespan normalized to the first iteration EDF schedule makespan. Note: searches starting from random schedules were unable to find feasible schedules.



(a) Change in makespan over time

(b) Aggregate improvement over baseline

Fig. 10. These figures depict benchmark results for 50 agents and 3 tasks, makespan normalized to the first iteration EDF schedule makespan. Note: searches starting from random schedules were unable to find feasible schedules.

First we note that in both scenarios, our baseline greedy condition of committing to one schedule yields results that match our observations from data collected in our human-subject pilot study. In the pilot study, we discovered that the natural learning rate for LEGO building tasks is rapid, indicating that any scheduling advantages must be captured in the initial rounds of scheduling. Across both scenarios, our simulated makespan for a fixed schedule decayed 35.1 $\pm$ 0.5 % across ten iterations, which is approximately the average amount of decay across repetitions of individual tasks in our pilot study. Thus we are confident that our simulation is realistic in terms of agent-task performance.

We see from Figure 9 and Figure 10 that layered application of our coordination algorithm components results in increasing returns on makespan efficiency. We then look at the breakdown of makespan improvement across iterations to see if we have met two objectives: 1) to explore to find a schedule that performs better than committing to the same schedule by iteration ten, where the original assignment would have resulted in agent task proficiency (i.e $< 2\%$ improvement by iteration ten), and 2) to avoid negative aggregate makespan efficiency during the exploration period, where assigning an agent with no experience to a task results in a large initial cost in efficiency. We summarize the average percent improvement of each condition and aggregate improvement across the finite horizon of ten iterations in Table 4 and 5.

Condition	Average Improvement	Aggregate Improvement	Final Round Improvement
Ours \ (λ and adaptation)	0.1 ± 0.5 %	0.6 ± 5.1 %	0.0 ± 0.8 %
Ours \ λ	4.0 ± 0.7 %	40.4 ± 6.9 %	4.7 ± 0.8 %
Ours	**4.5** ± **0.7** %	**44.8** ± **6.4** %	**5.1** ± **0.7 %**

Table 4. Schedule makespan percent improvement over baseline across iterations for 20 tasks, with standard error at $\alpha = 0.05$.

Condition	Average Improvement	Aggregate Improvement	Final Round Improvement
Ours \ (λ and adaptation)	-0.27 ± 0.3 %	-2.7 ± 3.2 %	-0.6 ± 0.5 %
Ours \ λ	1.5 ± 0.5 %	15.0 ± 4.9 %	1.9 ± 0.6 %
Adaptive PBO w/ lambda	**1.9** ± **0.6** %	**18.9** ± **5.5** %	**2.1** ± **0.7 %**

Table 5. Schedule makespan percent improvement over baseline across iterations for 50 tasks, with standard error at $\alpha = 0.05$.

Based on our empirical results, we find that full application of our coordination algorithm resulted in the highest aggregate improvement ($40.4 \pm 6.9\%$) across the first ten iterations, ending with an average improvement of $5.1 \pm 0.7\%$ on iteration ten as shown in Table 4. A net positive on iteration ten indicates that our schedule optimization yields better results than committing to the the same schedule until agents have achieved task proficiency. In addition, the large positive aggregate improvement over the ten iterations show that it is safe for the manufacturer to explore task assignments using our method without sacrificing net efficiency.

For 50 tasks, we saw full application of our coordination algorithm yielded the highest aggregate improvement at $18.9 \pm 5.5\%$ across the first ten iterations, as shown in Table 5. The results still show significant improvement over the baseline, albeit at a lower rate than coordination with 20 tasks. This reflects the increased difficulty in substitute sub-optimal agents for exploration of task-proficiency, as inter-task dependencies propagate delays downstream for the remainder of the schedule.

These results are valuable as they represent permanent increases in the schedule efficiency that could be autonomously applied to improve tight profit margins in real-world manufacturing settings. It is the combination of evolutionary optimization and adaptive learning curve updates that yields significant results over greedy scheduling. Lastly, across 50 randomly generated scheduling problems, a general purpose exploration coefficient of lambda = 0.1*starting makespan was able to equal performance in 0 scenarios and improve performance in 0 scenarios. λ can be easily tuned for each individual scheduling problem, based on expectations built from prior agent data. We weight our λ coefficient accordingly in our subsequent human-subjects experiment.

9.6 Key Takeaways

Our simulated scheduling problems for human-robot teams with stochastic and dynamic task proficiency validates:

1. Our upper-bound generation method allows for a fast, conservative bound while keeping computational costs near-negligible for a large number of tasks and only introducing a constant, low proportional rate of conservatism.
2. Our parametric updates to agent-specific learning curves consistently yield a better estimation of future task estimations. Thus we can improve our understanding of individual characters and task competency as we continually adapt our schedule.
3. Our comparison of search methods indicate that evolutionary-based search is a viable approach for schedule optimization.
4. Our benchmark study on the sequential inclusion of our coordination algorithm components indicate that we can effectively improve on greedy scheduling, despite the rapid advancements humans exhibit in learning new tasks.
5. Most importantly, these search methods can operate in real-time and are consistently capable of leveraging the information provided by rapid new insights into agent-specific capabilities.

In the following section, we apply the entirety of our algorithm in a human subjects study and validate our approach in factoring in trust and team fluency while also maximizing team efficiency.

10 USER STUDY OF HUMAN-ROBOT TEAMING IN AN ANALOGUE MANUFACTURING ENVIRONMENT

We conduct a between-subjects user study to gain insight into how a robotic scheduler should balance between exploration and exploitation strategies for effective human-robot collaboration in an analogous manufacturing setting. We recruited 96 participants from the university campus through mailing lists. The study involved 48 trials, with two subjects interacting with a robotic assistant to complete assembling multiple task schedules in each trial. Three trials were excluded from the analysis as the subjects failed to complete the entire schedule. The resulting 90 subjects consisted of 41 male and 31 female (18 subjects declined to provide their age and gender), ranged in age 18 to 32 (median: 23). None of the participants in this study had taken part in the pilot study. All participants received a $15 Amazon gift card as compensation for participating in the study. We obtained approval for human subject experimentation from the Institutional Review Board (Protocol H19258) before starting the study.

10.1 Task Design

To create an analogue environment of a final assembly line in a manufacturing plant, we assign the subjects to assemble LEGO pieces delivered by the robot at an assembly station. We chose LEGO as the assembly task based on prior user studies in Human-Robot collaboration in the context of manufacturing [31, 57]. For our study, we selected six assembly tasks from the LEGO 75261 Clone Scout Walker kit (Figure 11). The tasks were designed to have varying complexity and take approximately two to four minutes each to assemble. The subjects have access to the user manual while assembling each task.

10.2 Human-Robot Team Composition

The human-robot assembly team consists of two human workers (study participants), a robotic assistant (Sawyer), and a human experimenter. In each study session, both workers must collaborate with Sawyer to complete assembling all six LEGO tasks for multiple rounds. Sawyer is in charge of scheduling tasks in each round and fetching appropriate part kits for the human teammates. The workers follow instructions on a manual to complete assembling the task. The experimenter is present to supervise the entire study session.

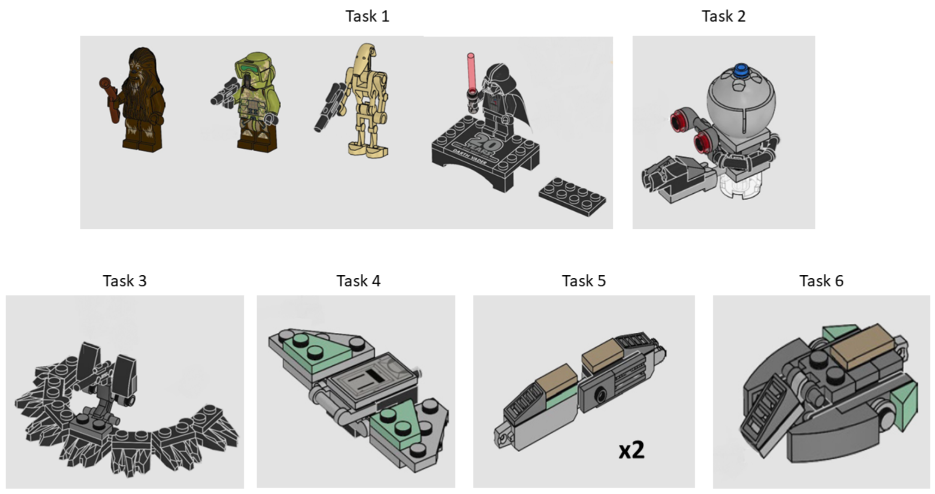


Fig. 11. This figure depicts the LEGO tasks used in the study (Image Courtesy: LEGO).

10.3 Scheduling Approaches

This study examines trends in schedule makespan, subject trust, and human-robot work alliance for different scheduling approaches in a LEGO assembly task. We propose three different exploitation-based strategies in this study and randomly assign participants to each condition with a balanced design. Scheduling strategy is an independent variable with three factor levels, each conforming to different values of λ in Equation 1:

- **Exploitation ($\lambda = 0$):** The algorithm seeks to minimize the schedule makespan based on prior task completion times.
- **Annealed Explore-Exploit (AEE, $\overline{\lambda} = 25$)**: The scheduler gradually decays the λ parameter to favor exploitation to a greater extent for the latter half of the session. The average λ annealing across five rounds is $\overline{\lambda} = 25$.
- **Explore-Exploit ($\lambda = 50$):** The algorithm is weighted towards considering novel assignments in the first half of the session by setting $\lambda = 50$ and tends to favor exploitation in the latter half of the strategy.

10.4 Experiment Procedure

10.4.1 Pre-Study. Upon arrival, the two participants are asked to sign separate consent forms, briefed about the study procedure and what is expected of them during the course of the study by the experimenter.

10.4.2 Study Setup. The setup consists of two tables representing the fetching area and assembling area, respectively. The robotic assistant Sawyer is situated in between the two tables. Initially, the fetching area contains all the LEGO kits to be assembled placed in separate bins. Both human participants are seated at the assembly area opposite the robot, as shown in Figure 1. There are two types of tasks involved in this experiment - 1) fetching part kits and 2) assembling part kits. The robot is capable of only fetching parts, while the human subjects assemble the tasks. The fetching task involves inspecting the bins and transporting them to the assembly area. The setup is intended to mimic a final assembly line in a manufacturing setting, which involves regular inspection of parts

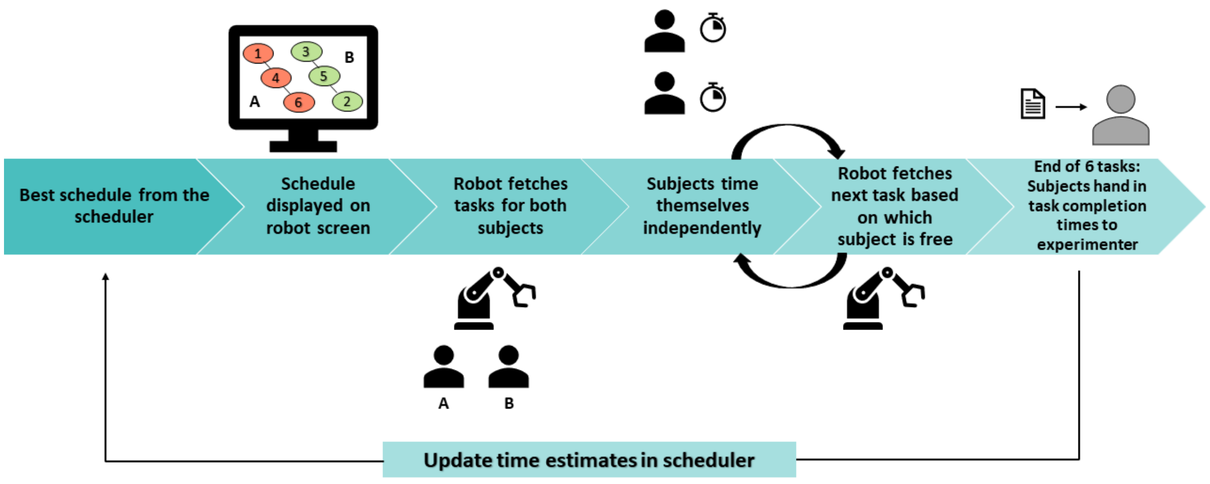

Best schedule from the scheduler
Schedule displayed on robot screen
Robot fetches tasks for both subjects
Subjects time themselves independently
Robot fetches next task based on which subject is free
End of 6 tasks: Subjects hand in task completion times to experimenter
Update time estimates in scheduler
A
B
1
3
4
5
6
2

Fig. 12. This figure depicts the study timeline for one round.

when transported from one zone to another. This setup is designed in accordance with prior work on human-robot collaboration on manufacturing lines [30, 31].

10.4.3 Study Session. After briefing the subjects, the experimenter picks a scheduling strategy at random for the current trial. This strategy is maintained constant throughout the trial. Each trial consists of five rounds of LEGO task building, and the schedule for each round is displayed to the subjects on Sawyer's screen. There are a total of six tasks to be completed in each schedule. The tasks in the schedule are color-coded differently for the two subjects. Each subject is required to complete three tasks in a schedule. The complexity and thereby, the completion times of individual tasks vary. The tasks in the first round are distributed among the subjects such that the total task completion time for the three tasks is approximately the same (based on task completion times in the pilot study). This task distribution is ensured to minimize the makespan, and is agnostic to the type of scheduling strategy chosen for the trial.

After computing the schedule for the current round, Sawyer proceeds to fetch the individual tasks for the two subjects and places them at the assembly area based on this schedule. The two subjects start each round of tasks concurrently and time themselves for the number of seconds they take to complete each task. The experimenter ensures that both the subjects start their respective timers together at the beginning of each round of assembly. For the remainder of the schedule, Sawyer fetches tasks based on which subject is available. For instance, the two participants A and B are assigned to complete tasks 1, 4, 6, and 3, 5, 2, as per the initial schedule. At the start of the round, Sawyer brings task bins 1 and 3 to the assembly area, and the two participants start assembling simultaneously. If subject B completes task 3 before A completes task 1, then Sawyer will fetch task 5 for B to avoid keeping B waiting. However, the subject has to wait if he/she completes a task while Sawyer is fetching the next task for the other teammate. This process continues until both subjects complete their three assigned tasks in each schedule.

At the end of each round (i.e., after completing the six tasks), the subjects hand over the task completion times to the experimenter, which are fed into Sawyer's algorithm to update the learning curve parameters of each subject. Sawyer then comes up with a different schedule for the next round based on the updated parameters and the scheduling approach chosen for the current trial. Finally, Sawyer comments on whether the tasks will be swapped or reassigned based on the previous round task completion times. For an exploration-based update, Sawyer would say, "We are going to try to mix it up and see how you both will perform. Here is the new schedule." For an exploitation-based update, Sawyer would say, "Based on the time you took in the previous round, I think this should be the new schedule."

The subjects participated in five rounds of LEGO building in a session, lasting for a total of about 45 minutes. The timeline for one round of the study is depicted in Figure 12.

10.4.4 Post Study. After completing five rounds of LEGO assembly, the participants are asked to complete a post-study questionnaire as detailed in Appendix A to assess their trust in the robot and their perceptions of team competence and robot performance. The participants are then debriefed about the purpose of the study.

10.5 Hypotheses

The fundamental proposition for our hypotheses is that exploration-based strategies will improve team performance and garner greater trust in the robot.

H3 *Schedule Makespan Improvement: Explore-Exploit ($\lambda = 50$) will have a more considerable improvement in schedule makespan, than pure exploitation or annealed explore-exploit.* As explore-exploit accounts for the highest entropy among all our scheduling approaches, we hypothesize that it will lead to maximum improvement in task completion times from the first round to the final round due to extensive search over different task allocation combinations. Schedule makespan improvement is the relative improvement in completion times of the succeeding rounds from the first round.

H4 *Robot Trust: The inclusion of some exploration in scheduling will engender greater trust.* Exploration favors different task allocations to the participants. As the subjects are exposed to varied task assignments in exploration-based strategies, they are cognizant of which tasks they are most proficient in and thereby have greater situational awareness about how the robot is trying to optimize their schedule. Prior studies in automation have shown how loss of situational awareness can negatively impact trust in the system [27, 30, 36]. Hence, we hypothesize that a better understanding of the scheduling process by subjects in exploration-based strategies will lead to greater trust in the robot. Robot trust is assessed using a 7-point Likert scale questionnaire adapted from [39].

H5 *Human-Robot Work Alliance: Perception of the team working alliance will increase in conditions with exploration in schedule assignments.* We hypothesize that exploration can lead to a greater understanding of task allocation per subject and thereby lead to an improved human-robot working alliance. Human-robot work alliance is calculated using a 7-point Likert scale of the working alliance questionnaire for human-robot teams from [39].

10.6 Metrics

In this section, we enlist both subjective and objective metrics used in evaluating the effectiveness of different scheduling methodologies for human-robot collaboration.

10.6.1 Objective Metrics. We conduct five rounds of task assembly, with an equal number of tasks allocated to the two human subject teammates in each round. We use the following objective metrics in our analysis of the different scheduling approaches.

M1 *Makespan Improvement:* Makespan is the total time taken to complete one schedule. As the participants work in parallel, and there are no task dependencies in the user study schedules, makespan is the maximum of the individual task completion times in each round. If τ_i^f represents the finish times for tasks in round k, then makespan for k^{th} round is given as $z_k = \max_{\tau_i \in \tau_k} \tau_i^f$.

The distribution of task times in the final round is dependent on both the scheduling strategy and the skill of the subjects in the assembly task at the start of the study. To evaluate the effectiveness of the scheduling strategies while being agnostic to the variations in the initial skill of the subjects in each session, we utilize relative improvement in makespans from the first round as our metric. We compare with respect to the first round, as the schedules in the first round are uniform across all strategies and are solely based on the task completion times collected from the pilot study. The average improvement in makespan for rounds two through five with respect to the first round makespan is considered as the makespan improvement metric and is calculated as

$$Makespan\ Improvement = \frac{1}{4} \sum_{k=2}^{5} 1 - \frac{z_k}{z_1} \tag{22}$$

M2 *Subject Skill:* Subject skill is a categorical variable with two levels - high and low. High skill denotes the participant with the shorter task completion time in every trial.

M3 *Subject Idle Time:* Subject idle time refers to the total time a subject had to spend waiting either for their human partner to complete assembling tasks, or for the robotic assistant Sawyer to fetch their respective after having completed their previously allocated task.

10.6.2 Subjective Metrics. We use 7-point Likert scale surveys to measure participant's perceptions of trust in the robot, human-robot work alliance, team competence, and robot performance. Table 6 in the Appendix shows the post-trial questionnaires used in the assessment of these metrics. We also report Cronbach's α for the internal validity of each of the questionnaires used in our analysis.

M4 *Robot Trust:* Trust in robots is a measurable attitude and is best reported using self-reported psychometric tests such as questionnaires. Robot trust is adapted from the trust questionnaire from [39].

M5 *Human-Robot Working Alliance:* Team fluency is crucial in determining the team's overall efficiency in the long run. This questionnaire measures the subject's view of how well the subject and the robot bonded together in achieving their overall objective. We utilize the working alliance questionnaire from [39].

M6 *Perceived Team Competence:* This questionnaire assesses the participant's satisfaction in overall team performance in each session.

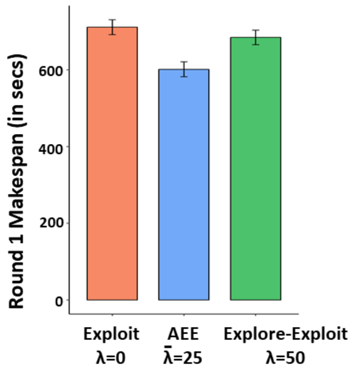
M7 *Perceived Robot Performance:* This questionnaire measures the participant's perception of the robot's commitment to the team and its overall performance in fetching relevant tasks.

M6 and **M7** are adaptions of subjective metrics used in [15] for human-robot teams with multiple human teammates.

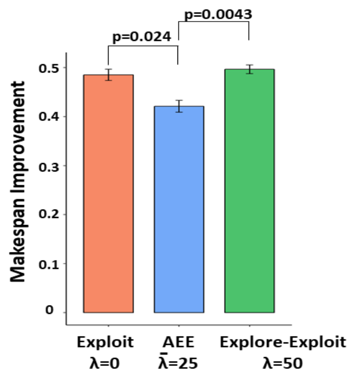
10.7 Study Results

In this section, we report the outcomes of deploying different scheduling approaches for human-robot collaboration in our user study in terms of trust, human-robot working alliance, and changes in human subject performance. The significance level for all our statistical analyses is set at $\alpha = 0.05$.

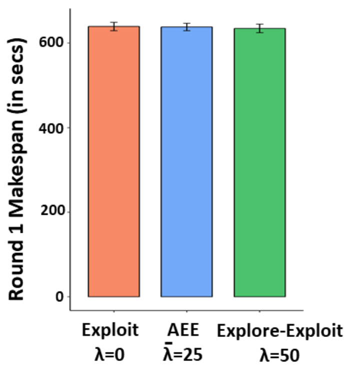
10.7.1 Analysis of H3 - Schedule Makespan Improvement. We first examine the effect of different strategies on the overall improvement in makespan across different rounds. Although the distribution of tasks in the first round



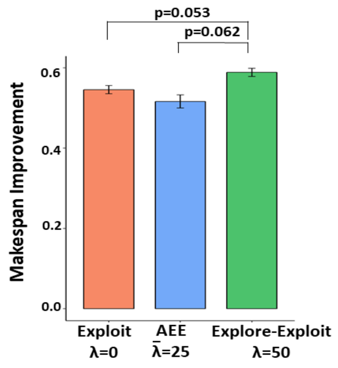
(a) Without Subject Matching



(b) Without Subject Matching



(c) With Subject Matching



(d) With Subject Matching

Fig. 13. This figure depicts comparisons of makespan improvement across different strategies with and without subject matching.

is agnostic to the scheduling strategy, the distribution of task completion times varies with the subject's familiarity with assembling LEGOs. As the subjects are assigned at random to each scheduling approach, the task completion times for the first round is not uniform (13a). Hence, we consider relative improvement in makespan with respect to first-round makespan. To evaluate the improvement in the performance of subjects on the assembly task across multiple iterations, we consider a linear model with the average improvement in makespan from rounds two through five as the dependent variable (**M1**) and strategy as the independent variable. As this model fails the normality of residuals assumption for one-way analysis of variance (ANOVA), we resort to a non-parametric test, namely Kruskal-Wallis, for our analysis. Results from Kruskal-Wallis shows that the schedule makespan improvement is highly significant on the type of scheduling strategy ($\chi^2 = 6.668$, $p = 0.00359$). Post-hoc analysis with Dunn's test shows that both explore-exploit ($\lambda = 50$) and pure exploitation ($\lambda = 0$) are significantly different from AEE ($\bar{\lambda} = 25$) with $p = 0.0043$, and $p = 0.024$ respectively as shown in Figure 13b. Trends indicate that makespan improvement is the highest for explore-exploit ($\lambda = 50$) and the least for AEE ($\bar{\lambda} = 25$).

Furthermore, since the learning curves of the human subjects follow a decaying exponential (**H1**), the subject's performance in the first round can impact the rate of improvement in performance in subsequent rounds. For a more robust assessment of the impact of the scheduling strategy on human subject performance, we perform nearest neighbor subject matching [71] based on makespans for the first round in R [66]. Post subject matching with first-round makespans, we consider 27 sessions from the initial distribution (9 sessions × 3 types of strategy = 27 sessions). We note that the average first-round makespans with subject matching are almost uniform (13c). We continue to observe maximum improvement in makespan for the explore-exploit condition ($\lambda = 50$) after subject matching 13d. Upon performing Kruskal-Wallis post subject matching, we once again find statistical significance for makespan and strategy($p = 0.03564$). Post-hoc analysis with Dunn's test reveals statistical significance between explore-exploit ($\lambda = 50$) and exploit ($\lambda = 0$), explore-exploit ($\lambda = 50$) and AEE ($\bar{\lambda} = 25$).

Takeaway: High entropy in the scheduling approach will lead to extensive search over task allocation combinations before exploiting. Consequently, we observe maximum improvement in makespan for explore-exploit ($\lambda = 50$), for conditions with and without subject matching. We postulate that the poor makespan improvement in AEE ($\bar{\lambda} = 25$) may be the effect of annealing λ, as this strategy tends to exploit before exploring sufficient novel task assignments.

10.7.2 Analysis of H4 - Robot Trust.

To analyze how subject's trust in the robot varies across strategies, we consider a linear model with trust as the dependent variable and strategy, subject skill (**M2**), average improvement in makespan (**M1**), perceived team competence (**M6**) and robot performance (**M7**) as independent variables. We first verified that this model satisfied the assumptions of ANOVA using Shapiro-Wilk's test ($p = 0.617$) for normality of residuals assumption and Levene's test for homoscedasticity ($p = 0.241$). Upon performing one-way ANOVA, we find that users' trust in the robot was not significant for the strategy used.

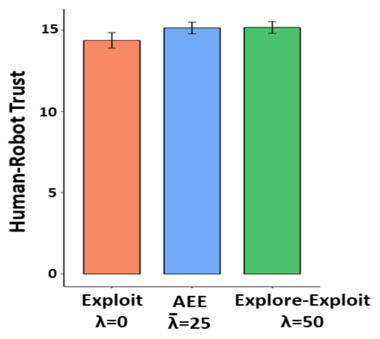
Trends indicate that strategies with more exploration yield higher trust, but the difference in means across strategies is not significant. Hence, we fail to reject the null hypothesis. However, trust is significantly dependent on the perceived team competence ($F(1, 90) = 10.63$, $p < 0.001$) and robot performance ($F(1, 90) = 18.54$, $p < 0.001$). This result indicates that trust in the robot depends directly on team and robot performance (Figure 14c) and robot performance (Figure 14d) rather than the scheduling approach.

Further, we analyzed the subjective responses of each pair of users based on their comparative performance in the assembly task. We find that the participants with longer task completion times in each session (low skill subjects **M2**), report a significant difference in trust when factoring λ as the independent variable ($F(2, 45) = 4.887$, $p = 0.0122$). We note that low skill users reported higher trust in the robot and higher perceived team competence for strategies involving exploration (Figure 14b).

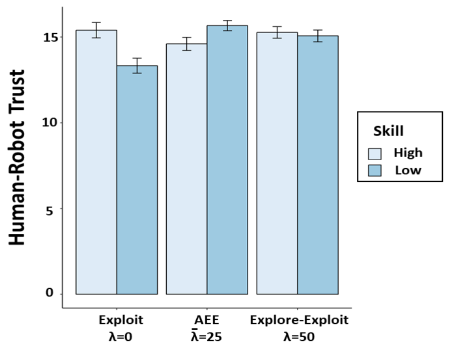
Takeaway: User's trust in the robot is statistically significant and positively correlated with other subjective metrics such as perceived team competence ($p < 0.001$), and perceived robot performance ($p < 0.001$). We also note that low-skill users report higher trust in exploration-based strategies than pure exploitation, as exploration-based strategies tend to balance out the workload after the first few iterations.

10.7.3 Analysis of H5 - Human-Robot Work Alliance.

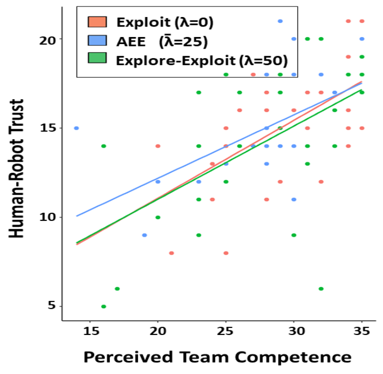
We examine the subject's perceptions of human-robot work alliance, by conducting ANOVA on a linear model with work-alliance (**M5**) as the dependent variable, and strategy, subject skill (**M2**), average improvement in makespan (**M1**), perceived team competence (**M6**) and robot performance (**M7**) as independent variables. We confirm that this model adheres to the assumptions of ANOVA by Shapiro-Wilk's Normality test ($p = 0.742$) and Levene's test for homoscedasticity ($p = 0.210$). We obtain statistical significance for scheduling strategy ($F(1, 90) = 3.843$, $p = 0.0251$) after ANOVA. Post-Hoc analysis with Tukey-HSD shows significance between explore-exploit ($\lambda = 50$) and exploit ($\lambda = 0$) strategies ($p = 0.0099$).



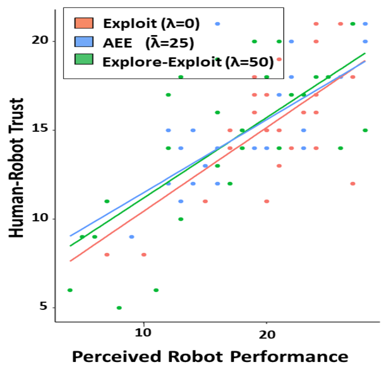
(a)



(b)

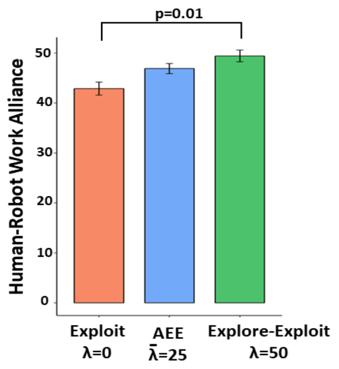


(c)

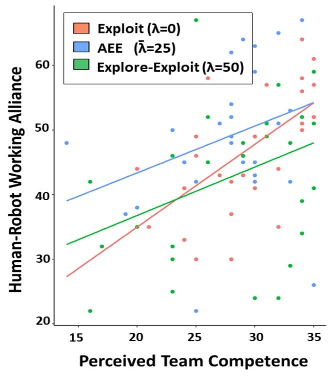


(d)

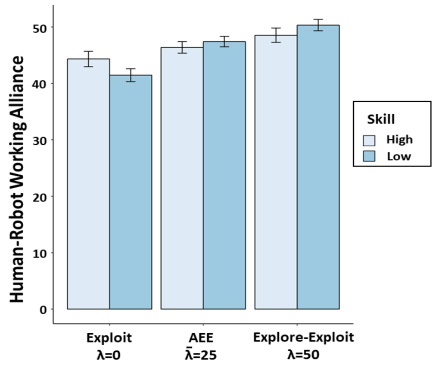
Fig. 14. Human-Robot Trust Analysis. (a) shows distribution of user's trust in the robot across scheduling strategies, (b) shows variation in user's trust categorized by strategy and speed, (c) and (d) represents how user's trust varies with subjective metrics - team competence and robot performance.



(a)



(b)



(c)

Fig. 15. Human-Robot Working Alliance Analysis. (a) shows the distribution of human-robot working alliance across different scheduling strategies. (b) represents the relation between the human-robot alliance and perceived team competence. (c) Human-robot working alliance categorized by strategy and skill.

Takeaway: Our statistical analysis shows that participant's perception of the human-robot work alliance favors strategies with exploration. Results from ANOVA also indicate that work alliance between humans and the robot is dependent on both subjective and objective metrics of team performance, i.e., perceived team competence (See Figure 15) ($F(1,90) = 21.289$, $p < 0.001$) and the average improvement in makespan ($F(1,90) = 8.218$, $p = 0.005$). We observe that both low-skill and high skill users' perceptions of human-robot working alliance increases with the exploration factor λ.

11 DISCUSSION

We conducted a user study in an analogue manufacturing setting to evaluate the effect of different exploitation-based strategies on human subject performance in an assembly task, while also assessing their perceptions of trust in the robotic assistant and the overall working alliance of the human-robot team. Our primary objective was to show that our coordination algorithm can improve objective and subjective team fluency by reasoning about schedule exploration and uncertainty. Our statistical analyses indicate that strategies that favor exploration to a greater extent lead to a significant improvement in human subject performance in terms of makespan ($p = 0.01$). The same strategies also result in higher team fluency as measured by human-robot working alliance ($p = 0.0035$).We observe a positive correlation between the user's perception of trust in the robot and subjective metrics of team performance ($p < 0.001$) and robot performance ($p < 0.001$). We also note that trust among low skill users is dependent on λ ($p = 0.012$) and tend to favor exploration-based strategies. Thus we find that one should adopt a scheduling mechanism that concurrently factors in exploration by accepting additional short-term costs to enable stronger long-run performance. We show both in simulation and in a physical HRI study that these methods are viable for aggregate makespan improvement, even in situations where the window of opportunity is narrow due to rapid learning for human task proficiency.

One of the limitations of our study was that our access to analogous manufacturing tasks was limited to observing LEGO tasks rather than people in real-world manufacturing. Additionally, fatigue is not a predominant factor in our study, which could be a consideration for a job lasting eight or more hours per day. Finally, our study was cross-sectional; a longitudinal study on learning effects and stochasticity in task performance for team coordination is warranted.

In future work, we would like to consider extensions to our algorithm that consider an active evolution policy, where candidate mutations are driven by information gain derived from the latest iteration of agent-task samples. We could also look to generate select mutation based on demonstrations by integrating human input on re-scheduling behavior. Such approaches could accelerate the discovery of stronger schedule candidates but must be weighted to counter misleading positive trends in task proficiency that are a result of natural task duration variance, rather than actual improvements via learning.

12 CONCLUSION

As advancements in robot capability progress, our machine counterparts have greater untapped opportunities to play a larger role in a joint human-robot workforce. Our objective is to foster the integration of human-robot teams, by enhancing the ability of the robot to explicitly reason about the capabilities of it's teammates. Prior work on behavioral teaming and the natural computational intractability of large-scale schedule optimization suggests that robots can offer a valuable service by designing and adapting schedules for robot and human teammates.

In this paper, we introduced an algorithm to enable robots to balance optimizing human perception of team working alliance with schedule efficiency by enabling exploration of candidate schedules while ensuring robustness to temporal constraints. We base our algorithm and corresponding model on two assumptions: that human's learn rapidly over time, and that as a population, human task completion times fall across a normal distribution. We validate our modeling assumptions by conducting a user study (n=18) in an analogue manufacturing environment where teammates assemble LEGO part kits for multiple iterations to analyze learning effects. From this user study, we create a prior distribution over human task performance, which forms the basis of further studies in virtual and real environments.

Through empirical evaluation, we showed our team coordination algorithm can reason about the future task duration for each individual agent's assignments by improving a model of that individual's learning curve, and use Gaussian upper-bounds to compute a fast evaluation of schedule robustness and makespan upper-bound. Combined, these approaches enable an online evolutionary-based search that can evaluate schedules quickly and change assignments in real-time to adapt to the needs and improve the efficacy of human-robot teams. Our simulation results show that our methods are applicable to scenarios where humans learn rapidly, and show significant net improvements ($p < 0.001$) over maintaining a myopic scheduler. In addition, we show that our methods work for schedules with a large number of tasks and inter-task temporal constraints, where previous methods for precise reasoning about uncertainty sets render the problem computationally intractable. Finally, results from human subjects experiment (n=90) support scheduling strategies that are inclusive of exploration for task allocation in human-robot teams as it engenders greater trust (among low performing individuals $p = 0.012$) and team working alliance ($p < 0.001$), while also generating improvements in schedule efficiency ($p = 0.00359$) through makespan reduction.

13 ACKNOWLEDGMENTS

This work was sponsored by ONR under grant N00014-18-S-B001, MIT Lincoln Laboratory grant 7000437192, and Georgia Tech institute funding.

REFERENCES

[1] S. Akhlaghi, N. Zhou, and Z. Huang. 2017. Adaptive adjustment of noise covariance in Kalman filter for dynamic state estimation. In *2017 IEEE Power Energy Society General Meeting*. 1–5. https://doi.org/10.1109/PESGM.2017.8273755

[2] M. Angélica Salazar-Aguilar, André Langevin, and Gilbert Laporte. 2014. The Multi-District Team Orienteering Problem. *Comput. Oper. Res.* 41 (Jan. 2014), 76–82. https://doi.org/10.1016/j.cor.2013.07.026

[3] Laura Barbulescu, Zachary Rubinstein, Stephen Smith, and Terry Zimmerman. 2010. Distributed coordination of mobile agent teams: the advantage of planning ahead. 1331–1338. https://doi.org/10.1145/1838206.1838380

[4] B. R. Barmish and C. M. Lagoa. 1996. The uniform distribution: rigorous justification for its use in robustness analysis. In *Proceedings of 35th IEEE Conference on Decision and Control*, Vol. 3. 3418–3423 vol.3.

[5] Aharon Ben-Tal and Arkadi Nemirovski. 2000. Robust solutions of Linear Programming problems contaminated with uncertain data. *Mathematical Programming* 88, 3 (01 Sep 2000), 411–424. https://doi.org/10.1007/PL00011380

[6] Aharon Ben-Tal and Arkadi Nemirovski. 2002. On Tractable Approximations of Uncertain Linear Matrix Inequalities Affected by Interval Uncertainty. *SIAM Journal on Optimization* 12 (01 2002), 811–833. https://doi.org/10.1137/S1052623400374756

[7] Dimitris Bertsimas and Melvyn Sim. 2004. The Price of Robustness. *Oper. Res.* 52, 1 (Jan. 2004), 35–53. https://doi.org/10.1287/opre.1030.0065

[8] James C. Boerkoel and Edmund H. Durfee. 2012. A Distributed Approach to Summarizing Spaces of Multiagent Schedules. In *AAAI*.

[9] James Boerkoel Jr and Edmund Durfee. 2013. Decoupling the Multiagent Disjunctive Temporal Problem. *Proceedings of the 27th AAAI Conference on Artificial Intelligence, AAAI 2013* 2.

[10] Giorgio C. Buttazzo. 2011. *Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications* (3rd ed.). Springer Publishing Company, Incorporated.

[11] G. C. Calafiore and M. C. Campi. 2006. The scenario approach to robust control design. *IEEE Trans. Automat. Control* 51, 5 (May 2006), 742–753. https://doi.org/10.1109/TAC.2006.875041

[12] G. C. Calafiore, F. Dabbene, and R. Tempo. 2000. Randomized algorithms for probabilistic robustness with real and complex structured uncertainty. *IEEE Trans. Automat. Control* 45, 12 (2000), 2218–2235.

[13] T. Chaari, S. Chaabane, N. Aissani, and D. Trentesaux. 2014. Scheduling under uncertainty: Survey and research directions. In *2014 International Conference on Advanced Logistics and Transport (ICALT)*. 229–234. https://doi.org/10.1109/ICAdLT.2014.6866316

[14] Tarek Chaari, Sondes Chaabane, Taïcir Loukil, and Damien Trentesaux. 2011. A genetic algorithm for robust hybrid flow shop scheduling. *Int. J. Computer Integrated Manufacturing* 24 (09 2011), 821–833. https://doi.org/10.1080/0951192X.2011.575181

[15] Crystal Chao and Andrea Thomaz. 2012. Timing in Multimodal Turn-Taking Interactions: Control and Analysis Using Timed Petri Nets. *Journal of Human-Robot Interaction* (08 2012), 4–25. https://doi.org/10.5898/JHRI.1.1.Chao

[16] A. Charnes and W. W. Cooper. 1959. Chance-Constrained Programming. *Management Science* 6, 1 (1959), 73–79. https://doi.org/10.1287/mnsc.6.1.73 arXiv:https://doi.org/10.1287/mnsc.6.1.73

[17] Min Chen, Stefanos Nikolaidis, Harold Soh, David Hsu, and Siddhartha Srinivasa. 2018. Planning with Trust for Human-Robot Collaboration. 307–315. https://doi.org/10.1145/3171221.3171264

[18] George Chryssolouris, Dimitris Mourtzis, and S. Makris. 2003. An Approach to Planning and Control of Ship Repair Manufacturing Operations. *CIRP Journal of Manufacturing Science and Technology* 32 (01 2003), 13–19.

[19] Houston Claure, Yifang Chen, Jignesh Modi, Malte Jung, and Stefanos Nikolaidis. 2019. Reinforcement Learning with Fairness Constraints for Resource Distribution in Human-Robot Teams.

[20] Amanda Coles, Andrew Coles, Maria Fox, and Derek Long. 2009. Incremental Constraint-Posting Algorithms in Interleaved Planning and Scheduling. *COPLAS 2009 - Proceedings of the Workshop on Constraint Satisfaction Techniques for Planning and Scheduling Problems* (01 2009).

[21] G. B. Dantzig and J. H. Ramser. 1959. The Truck Dispatching Problem. *Manage. Sci.* 6, 1 (Oct. 1959), 80–91. https://doi.org/10.1287/mnsc.6.1.80

[22] Rina Dechter and Judea Pearl. 1987. Network-based heuristics for constraint-satisfaction problems. *Artificial Intelligence* 34, 1 (1987), 1 – 38. https://doi.org/10.1016/0004-3702(87)90002-6

[23] M. Desai, P. Kaniarasu, M. Medvedev, A. Steinfeld, and H. Yanco. 2013. Impact of robot failures and feedback on real-time trust. In *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. 251–258. https://doi.org/10.1109/HRI.2013.6483596

[24] M. B. Dias, R. Zlot, N. Kalra, and A. Stentz. 2006. Market-Based Multirobot Coordination: A Survey and Analysis. *Proc. IEEE* 94, 7 (2006), 1257–1270.

[25] L. El Ghaoui and H. Lebret. 1997. Robust Solutions to Least-Squares Problems with Uncertain Data. *SIAM J. Matrix Anal. Appl.* 18, 4 (1997), 1035–1064. https://doi.org/10.1137/S0895479896298130 arXiv:https://doi.org/10.1137/S0895479896298130

[26] Laurent El Ghaoui, Francois Oustry, and Hervé Lebret. 1998. Robust Solutions to Uncertain Semidefinite Programs. *SIAM Journal on Optimization* 9, 1 (1998), 33–52. https://doi.org/10.1137/S1052623496305717 arXiv:https://doi.org/10.1137/S1052623496305717

[27] Mica Endsley. 1995. Endsley, M.R.: Toward a Theory of Situation Awareness in Dynamic Systems. Human Factors Journal 37(1), 32-64. *Human Factors: The Journal of the Human Factors and Ergonomics Society* 37 (03 1995), 32–64. https://doi.org/10.1518/001872095779049543

[28] Cheng Fang, Peng Yu, and Brian C. Williams. 2014. Chance-Constrained Probabilistic Simple Temporal Problems. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence* (Québec City, Québec, Canada) *(AAAI'14)*. AAAI Press, 2264–2270.

[29] Ernst Fehr and Armin Falk. 2002. Psychological Foundations of Incentives. *European Economic Review* 46 (01 2002), 687–724. https://doi.org/10.1016/S0014-2921(01)00208-2

[30] Matthew Gombolay, Anna Bair, Cindy Huang, and Julie Shah. 2017. Computational design of mixed-initiative human–robot teaming that considers human factors: situational awareness, workload, and workflow preferences. *The International Journal of Robotics Research* 36 (02 2017), 027836491668825. https://doi.org/10.1177/0278364916688255

[31] Matthew C. Gombolay, Reymundo A. Gutierrez, Shanelle G. Clarke, Giancarlo F. Sturla, and Julie A. Shah. 2015. Decision-making authority, team efficiency and human worker satisfaction in mixed human–robot teams. *Autonomous Robots* 39, 3 (01 Oct 2015), 293–312. https://doi.org/10.1007/s10514-015-9457-9

[32] Matthew Craig Gombolay, Cindy Huang, and Julie A. Shah. 2015. Coordination of Human-Robot Teaming with Human Task Preferences. In *AAAI Fall Symposia*.

[33] Matthew C. Gombolay, Reed Jensen, Jessica Stigile, Toni Golen, Neel Shah, Sung-Hyun Son, and Julie A. Shah. 2018. Human-Machine Collaborative Optimization via Apprenticeship Scheduling. *CoRR* abs/1805.04220 (2018). arXiv:1805.04220 http://arxiv.org/abs/1805.04220

[34] M. C. Gombolay, R. J. Wilcox, and J. A. Shah. 2018. Fast Scheduling of Robot Teams Performing Tasks With Temporospatial Constraints. *IEEE Transactions on Robotics* 34, 1 (Feb 2018), 220–239. https://doi.org/10.1109/TRO.2018.2795034

[35] Bram L. Gorissen, İhsan Yanıkoğlu, and Dick den Hertog. 2015. A practical guide to robust optimization. *Omega* 53 (2015), 124 – 137. https://doi.org/10.1016/j.omega.2014.12.006

[36] David Grimm, Mustafa Demir, Jamie Gormana, and Nancy Cookeb. 2018. Team Situation Awareness in Human-Autonomy Teaming: A Systems Level Approach. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 62, 149–149. https://doi.org/10.1177/1541931218621034

[37] Victoria Groom and Clifford Nass. 2007. Can robots be teammates? Benchmarks in human-robot teams. *Interaction Studies* 8 (10 2007), 483–500. https://doi.org/10.1075/is.8.3.10gro

[38] Willy Herroelen and Roel Leus. 2005. Project scheduling under uncertainty: Survey and research potentials. *European Journal of Operational Research* 165, 2 (2005), 289 – 306. https://doi.org/10.1016/j.ejor.2004.04.002 Project Management and Scheduling.

[39] G. Hoffman. 2019. Evaluating Fluency in Human–Robot Collaboration. *IEEE Transactions on Human-Machine Systems* 49, 3 (June 2019), 209–218. https://doi.org/10.1109/THMS.2019.2904558

[40] Chien-Ming Huang, Maya Cakmak, and Bilge Mutlu. 2015. Adaptive Coordination Strategies for Human-Robot Handovers. https://doi.org/10.15607/RSS.2015.XI.031

[41] L. Hunsberger and B. J. Grosz. 2000. A combinatorial auction for collaborative planning. In *Proceedings Fourth International Conference on MultiAgent Systems*. 151–158.

[42] Mikkel T. Jensen. 2001. Improving robustness and flexibility of tardiness and total flow-time job shops using robustness measures. *Applied Soft Computing* 1, 1 (2001), 35 – 52. https://doi.org/10.1016/S1568-4946(01)00005-9

[43] P. Khargonekar and A. Tikku. 1996. Randomized algorithms for robust control analysis and synthesis have polynomial complexity. In *Proceedings of 35th IEEE Conference on Decision and Control*, Vol. 3. 3470–3475 vol.3.

[44] Sven Koenig, Craig Tovey, Michail Lagoudakis, Evangelos Markakis, David Kempe, Pinar Keskinocak, Anton Kleywegt, Adam Meyerson, and Sonal Jain. 2006. The Power of Sequential Single-Item Auctions for Agent Coordination.

[45] G. Ayorkor Korsah, Balajee Kannan, Brett Browning, and M. Bernardine Dias. 2011. *xBots: An Approach to Generating and Executing Optimal Multi-Robot Plans with Constraints*. Technical Report CMU-RI-TR-11-25. Carnegie Mellon University, Pittsburgh, PA.

[46] Woo Young Kwon and Il Hong Suh. 2014. Planning of Proactive Behaviors for Human–Robot Cooperative Tasks under Uncertainty. *Knowledge-Based Systems* 72 (12 2014). https://doi.org/10.1016/j.knosys.2014.08.021

[47] Michail G. Lagoudakis, Evangelos Markakis, David Kempe, Pinar Keskinocak, Anton J. Kleywegt, Sven Koenig, Craig A. Tovey, Adam Meyerson, and Sonal Jain. 2005. Auction-Based Multi-Robot Routing. In *Robotics: Science and Systems*.

[48] Michael Lewis, Katia Sycara, and Phillip Walker. 2018. *The Role of Trust in Human-Robot Interaction*. Springer International Publishing, Cham, 135–159. https://doi.org/10.1007/978-3-319-64816-3_8

[49] Alan S. Manne. 1960. On the Job-Shop Scheduling Problem. *Operations Research* 8, 2 (1960), 219–223. https://doi.org/10.1287/opre.8.2.219 arXiv:https://doi.org/10.1287/opre.8.2.219

[50] Masahiro Ono and B. C. Williams. 2008. Iterative Risk Allocation: A new approach to robust Model Predictive Control with a joint chance constraint. In *2008 47th IEEE Conference on Decision and Control*. 3427–3432. https://doi.org/10.1109/CDC.2008.4739221

[51] Maitreyi Nanjanath and Maria Gini. 2010. Repeated auctions for robust task execution by a robot team. *Robotics and Autonomous Systems* 58, 7 (2010), 900 – 909. https://doi.org/10.1016/j.robot.2010.03.011 Advances in Autonomous Robots for Service and Entertainment.

[52] Arkadi Nemirovski and Alexander Shapiro. 2006. Convex Approximations of Chance Constrained Programs. *SIAM Journal on Optimization* 17 (2006), 969–996.

[53] S. Nikolaidis and J. Shah. 2013. Human-robot cross-training: Computational formulation, modeling and evaluation of a human team training strategy. In *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. 33–40. https://doi.org/10.1109/HRI.2013.6483499

[54] Ernesto Nunes and Maria Gini. 2015. Multi-robot Auctions for Allocation of Tasks with Temporal Constraints. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence* (Austin, Texas) *(AAAI'15)*. AAAI Press, 2110–2216. http://dl.acm.org/citation.cfm?id=2886521.2886614

[55] Ernesto Nunes, Marie D. Manner, Hakim Mitiche, and Maria L. Gini. 2017. A taxonomy for task allocation problems with temporal and ordering constraints. *Robotics and Autonomous Systems* 90 (2017), 55–70.

[56] Léon Planken, Mathijs Weerdt, and Roman Krogt. 2008. P3C: A New Algorithm for the Simple Temporal Problem. *Sigmod Record*, 256–263.

[57] S.M. Rahman and Yue Wang. 2018. Mutual trust-based subtask allocation for human–robot collaboration in flexible lightweight assembly in manufacturing. *Mechatronics* 54 (10 2018), 94–109. https://doi.org/10.1016/j.mechatronics.2018.07.007

[58] Ramya Ramakrishnan, Chongjie Zhang, and Julie Shah. 2017. Perturbation Training for Human-Robot Teams. *Journal of Artificial Intelligence Research* 59 (07 2017), 495–541. https://doi.org/10.1613/jair.5390

[59] F.E. Ritter and L.J. Schooler. 2001. *Learning Curve, The*. 8602–8605. https://doi.org/10.1016/B0-08-043076-7/01480-7

[60] Paul Robinette, Ayanna Howard, and Alan Wagner. 2017. Effect of Robot Performance on Human–Robot Trust in Time-Critical Situations. *IEEE Transactions on Human-Machine Systems* PP (01 2017), 1–12. https://doi.org/10.1109/THMS.2017.2648849

[61] N. Roy, P. Newman, and S. Srinivasa. 2013. *Optimization of Temporal Dynamics for Adaptive Human-Robot Interaction in Assembly Manufacturing*. MITP. https://ieeexplore.ieee.org/document/6577926

[62] I. Sabuncuoglu and S. Goren. 2009. Hedging production schedules against uncertainty in manufacturing environment with a review of robustness and stability research. *International Journal of Computer Integrated Manufacturing* 22, 2 (2009), 138–157. https://doi.org/10.1080/09511920802209033 arXiv:https://doi.org/10.1080/09511920802209033

[63] Maha Salem, Gabriella Lakatos, Farshid Amirabdollahian, and Kerstin Dautenhahn. 2015. Would You Trust a (Faulty) Robot?: Effects of Error, Task Type and Personality on Human-Robot Cooperation and Trust. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction* (Portland, Oregon, USA) *(HRI '15)*. ACM, New York, NY, USA, 141–148. https://doi.org/10.1145/2696454.2696497

[64] C. W. Scherer. 2005. Relaxations for Robust Linear Matrix Inequality Problems with Verifications for Exactness. *SIAM J. Matrix Anal. Appl.* 27, 2 (2005), 365–395. https://doi.org/10.1137/S0895479803430953 arXiv:https://doi.org/10.1137/S0895479803430953

[65] Sarah Sebo, Ling Dong, Nicholas Chang, and Brian Scassellati. 2020. Strategies for the Inclusion of Human Members within Human-Robot Teams. 309–317. https://doi.org/10.1145/3319502.3374808

[66] Jasjeet S. Sekhon. 2011. Multivariate and Propensity Score Matching Software with Automated Balance Optimization: The Matching Package for R. *Journal of Statistical Software* 42, 7 (2011), 1–52. http://www.jstatsoft.org/v42/i07/

[67] Marc Sevaux and Kenneth Sörensen. 2004. A genetic algorithm for robust schedules in a just-in-time environment with ready times and due dates. (01 2004).

[68] J. Shah, J. Wiken, B. Williams, and C. Breazeal. 2011. Improved human-robot team performance using Chaski, A human-inspired plan execution system. In *2011 6th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. 29–36. https://doi.org/10.1145/1957656.1957668

[69] A. L. Soyster. 1973. Convex Programming with Set-Inclusive Constraints and Applications to Inexact Linear Programming. *Operations Research* 21, 5 (1973), 1154–1157. http://www.jstor.org/stable/168933

[70] R. F. Stengel and L. E. Ryan. 1991. Stochastic robustness of linear time-invariant control systems. *IEEE Trans. Automat. Control* 36, 1 (1991), 82–87.

[71] Elizabeth Stuart. 2010. Matching Methods for Causal Inference: A Review and a Look Forward. *Statistical science : a review journal of the Institute of Mathematical Statistics* 25 (02 2010), 1–21. https://doi.org/10.1214/09-STS313

[72] V. Suresh and Dipak Chaudhuri. 1993. Dynamic scheduling—a survey of research. *International Journal of Production Economics* 32, 1 (1993), 53 – 63. https://doi.org/10.1016/0925-5273(93)90007-8

[73] Ioannis Tsamardinos. 2002. A Probabilistic Approach to Robust Execution of Temporal Plans with Uncertainty, Vol. 2308. 97–108. https://doi.org/10.1007/3-540-46014-4_10

[74] Panagiota Tsarouchi, Alexandros-Stereos Matthaiakis, Sotiris Makris, and George Chryssolouris. 2017. On a human-robot collaboration in an assembly cell. *International Journal of Computer Integrated Manufacturing* 30, 6 (2017), 580–589. https://doi.org/10.1080/0951192X.2016.1187297 arXiv:https://doi.org/10.1080/0951192X.2016.1187297

[75] Vladimir N. Vapnik. 1998. *Statistical Learning Theory*. Wiley-Interscience.

[76] M. Vidyasagar. 1997. *A Theory of Learning and Generalization: With Applications to Neural Networks and Control Systems*. Springer-Verlag, Berlin, Heidelberg.

[77] M. Vidyasagar. 2001. Randomized algorithms for robust controller synthesis using statistical learning theory. *Automatica* 37, 10 (2001), 1515 – 1528. https://doi.org/10.1016/S0005-1098(01)00122-4

[78] T. Williams, D. Szafir, T. Chakraborti, and E. Phillips. 2019. Virtual, Augmented, and Mixed Reality for Human-Robot Interaction (VAM-HRI). In *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. 671–672.

[79] Yang Xu, Paul Scerri, Bin Yu, Steven Okamoto, and Michael Lewis. 2005. An integrated token-based algorithm for scalable coordination. 407–414. https://doi.org/10.1145/1082473.1082536

[80] X. Jessie Yang, Vaibhav V. Unhelkar, Kevin Li, and Julie A. Shah. 2017. Evaluating Effects of User Experience and System Transparency on Trust in Automation. In *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction* (Vienna, Austria) *(HRI '17)*. Association for Computing Machinery, New York, NY, USA, 408–416. https://doi.org/10.1145/2909824.3020230

A STUDY QUESTIONNAIRE

Post-trial Questionnaire
Robot Trust ($\alpha = 0.70746$)
1. The robot was trustworthy.
2. The robot and I trusted each other.
3. I trusted Sawyer's decisions
Working alliance for human-robot teams ($\alpha = 0.8180$)
4. I feel uncomfortable with the robot (**reverse scale**).
5. The robot and I understand each other.
6. I believe the robot likes me.
7. The robot and I respect each other.
8. I feel that the robot appreciates me.
9. The robot and I trust each other.
10. The robot worker perceives accurately what my goals are.
11. The robot worker does not understand what I am trying to accomplish.
12. The robot and I are working towards mutually agreed upon goals.
13. I find what I am doing with the robot confusing (**reverse scale**).
Perceived Team Competence ($\alpha = 0.7356$)
14. I was satisfied by the team's performance.
15. The team collaborated well together.
16. The task performed tasks in the least time possible.
17. The task was difficult for us to complete together (**reverse scale**).
18. We were efficient in completing the tasks.
Perceived Robot Performance ($\alpha = 0.8023$)
19. The robot increased the productivity of the team.
20. The robot worker was necessary for the successful completion of the tasks.
21. Sawyer was team-oriented.
22. Sawyer's performance was important for completing the tasks.

Table 6. Post trial questionnaire