

Bilevel Optimization for Just-in-Time Robotic Kitting and Delivery via Adaptive Task Segmentation and Scheduling

Yi-Shiuan Tung, Kayleigh Bishop, Bradley Hayes, and Alessandro Roncone

Abstract— Kitting refers to the task of preparing and grouping necessary parts and tools (or “kits”) for assembly in a manufacturing environment. Automating this process simplifies the assembly task for human workers and improves efficiency. Existing automated kitting systems adhere to scripted instructions and predefined heuristics. However, given variability in the availability of parts and logistic delays, the inflexibility of existing systems can limit the overall efficiency of an assembly line. In this paper, we propose a bilevel optimization framework to enable a robot to perform task segmentation-based part selection, kit arrangement, and delivery scheduling to provide custom-tailored kits *just in time*—i.e., right when they are needed. We evaluate the proposed approach both through a human subjects study ($n=18$) involving the construction of a flat-pack furniture table and shop-flow simulation based on the data from the study. Our results show that the just-in-time kitting system is objectively more efficient, resilient to upstream shop flow delays, and subjectively more preferable as compared to baseline approaches of using kits defined by rigid task segmentation boundaries defined by the task graph itself or a single kit that includes all parts necessary to assemble a single unit.

I. INTRODUCTION

In the conventional or “single model” assembly line, only one product type is manufactured at a time. While this traditional manufacturing approach simplifies the responsibilities of line workers, it limits flexibility: making one product at a time means costly changeovers when switching product types, making it difficult to respond fluidly to changing dynamics of customer demand, upstream part shortages, and customizations. For this reason, manufacturers may employ “mixed model” assembly, in which multiple products — with different parts and assembly steps — are assembled on the same line. This approach smooths upstream part demand and can lead to much greater overall productivity in the assembly line [23].

The trade-off of mixed model assembly is that it requires assembly line workers to take on a greater range of tasks. Rather than constantly performing a rote series of assembly steps, workers now need to keep track of the various parts and steps for each product on the line. One approach to streamlining this step at the worker level is *kitting*. In manufacturing systems, kitting refers to the process of collecting components in a “kit” or container before feeding them to workstations where intermediate or end products are built. Kitting possesses a number of advantages: it 1) reduces

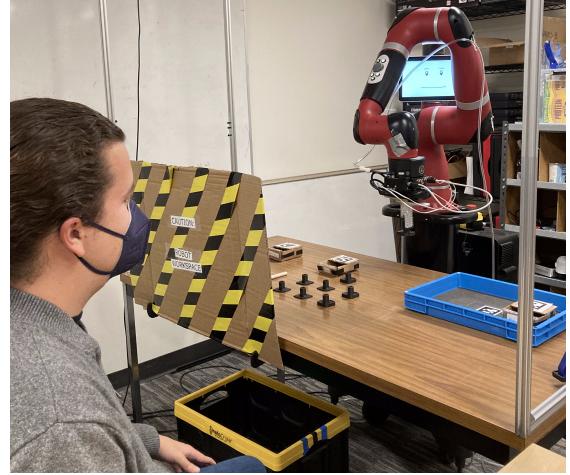


Fig. 1: This work presents an optimization approach to automate just-in-time robot part delivery for collaborative assembly. Here, a robot gathers parts required for an assembly task and places them on a kitting tray. The human then uses this “kit” to complete the next set of tasks for continuing to assemble the product.

storage space requirements at the workstations, 2) enables efficient product changeover (since common components are stored at a central location), and 3) increases worker productivity by reducing time spent gathering parts [2], [7].

Kitting is an excellent opportunity space for leveraging collaborative robotics to support and improve the efficiency of workers. While automation has seen great success in many areas of manufacturing, assembly tasks often remain too complex to automate completely: robots lack the fine-motor capabilities required for dexterous manipulation of small parts and tools. Kitting, however, is well within the mechanical capabilities of modern robots, and its automation would allow a robot to prepare kits while more dexterous agents assemble the product. This parallelism could greatly decrease the time and effort required to complete each product assembly.

In this work we propose and evaluate a dynamic robotic kitting planner that segments and schedules assembly tasks to minimize idle time and reduce makespan. The primary contributions of this work are: a) a novel optimization algorithm for just-in-time robotic kitting, which incorporates a model of the assembly goal, estimates of human and robot task times, and estimated usability of the generated kit layout; b) a demonstration of the optimization and its performance

*This work was supported by the Army Research Laboratory under grant nos. W911NF-20-2-0083, W911NF-21-2-0290, and W911NF-21-2-0123.

The authors are affiliated with the Department of Computer Science, University of Colorado Boulder, 1111 Engineering Drive, Boulder, CO 80309, U.S.A. {firstname}.lastname@colorado.edu

outcomes on an experimental human-robot collaboration scenario; and c) an analysis of the optimization’s robustness to logistic delays in simulation.

We formulate just in time kitting and delivery as a *bilevel* optimization, where the “upper-level” problem of task scheduling and segmentation is optimized in concert with the “lower-level” problem of part kitting. By performing this nested optimization, the robot adapts to its kitting conditions (e.g. kit arrangement and part availability) while syncing its workflow to that of the human.

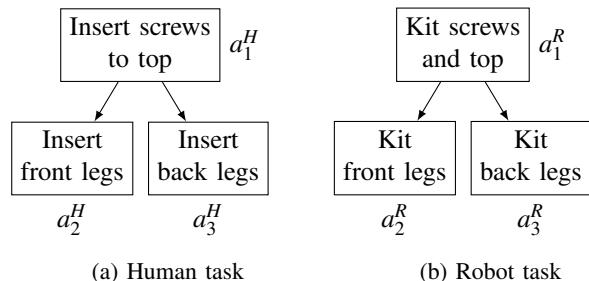
To evaluate our kitting approach’s ability to increase efficiency and improve users’ experience in a human-robot collaborative assembly, we implement our system on a Sawyer robot (Figure 1) and conduct a within-subjects user study of collaborative furniture assembly. To evaluate the system’s performance in scenarios with logistic delays, we use data from the user study to simulate the assembly process under varying conditions of part availability and throughput to explore our system’s robustness.

II. BACKGROUND & RELATED WORK

One of the disadvantages of kitting is that the preparation of kits consumes worker time and effort that don’t directly contribute to the assembly task. Kitting in mixed-model assembly lines also has higher rates of error [11] and assembly workers report higher cognitive loads [12], [17]. Due to the physical and cognitive demands of manual kit preparation, robotic kitting has been explored to increase efficiency and flexibility. Caputo et al. [3] developed an economic model to show that automated systems are more cost effective and efficient, except in low production settings. Prior work has also demonstrated the promise of human-robot collaborative kitting scenarios. Boudella et al. [1] developed a delivery-time-sensitive model to optimally distribute kitting tasks between a human and robot in a mixed-model assembly line, and saw significant efficiency improvements over baselines where the human worker took on the majority of the tasks. A mixed-integer linear programming approach to online kit delivery scheduling is presented in [14], in which the robot takes over all kitting tasks and results in large productivity improvements over a manual kitting baseline.

Our work differs from the above examples in that it presents a fully-automated kitting system that achieves Just in Time (JIT) part feeding, a lean manufacturing principle designed to eliminate waste in production [7]. The goal of JIT manufacturing is to feed the required quantity of parts to the workstations when they are needed, rather than before or after. Traditionally, JIT kitting is accomplished by determining the required kits at each workstation and designing the assembly layouts and kit sizes accordingly [18]. Other production systems use material handling techniques like kitting trolleys [15] or automated delivery vehicles [22] to achieve JIT.

In this work, we use task segmentation and scheduling of kitting tasks to achieve JIT delivery. Our work builds on progress in robot kitting by presenting a bilevel optimization approach to maximize the efficiency of the assembly process.



(a) Human task (b) Robot task

Fig. 2: A task graph for assembly of a stool. The robot retrieves the parts required for the human to complete the assembly. The arrows indicate precedence constraints. The robot tasks are represented by $T^R = \{a_2^R, a_3^R\}$ and the human tasks are represented by $T^H = \{a_1^H, a_2^H, a_3^H\}$.

By incorporating an awareness of human and robot task times, as well as part availability into kit layout and design, our system is more capable of adapting its kitting behavior to meet the changing demands of mixed model assembly lines.

III. PROBLEM FORMULATION

Consider a two-station, human-robot collaborative task where a human assembler works at an assembly workstation and a robot prepares kits at a preparation workstation. The robot receives part deliveries and produces kits, while the human receives kits and produces product. Let:

$$\mathcal{A} = \{a_1, a_2, \dots, a_n\}$$

be the set of n assembly tasks that need to be performed to complete the assembly. Each task a_i is composed of the robot-dependent kitting task a_i^R , in which the robot gathers the necessary parts, and the human-dependent assembly task a_i^H . The robot has to complete preparation task a_i^R (i.e., kitting and delivering components) before the human can begin a_i^H (assembling). In order to schedule its actions, the robot’s goal is to compute a *task ordering* over \mathcal{A} for itself and the human to follow in order to finish the assembly as quickly as possible.

The tasks in \mathcal{A} may not need to be completed in a strict order, but some tasks may have ordering constraints — for example, one may need to complete “insert screws to top” before one can perform “insert front leg” to build a table. Let us say that if task a_i must be completed before a_j , then a_i has a *precedence constraint* with a_j . Any task ordering the robot computes must satisfy these precedence constraints. See Figure 2 for an example task precedence graph, and the associated part retrievals the robot needs to perform.

To create a *schedule* to complete each task given a task ordering, the robot *segments* tasks into groups. This means the robot can prepare a kit one segment at a time, rather than having to make one kit for each of the human’s tasks. This also introduces the additional constraint that a segment can only be valid if all the associated parts can fit in the kit. Let us represent a task segment at time step t as $T_t \subseteq \mathcal{A}$.

Once the tasks have been segmented into groups, the robot generates a schedule by determining when to start each

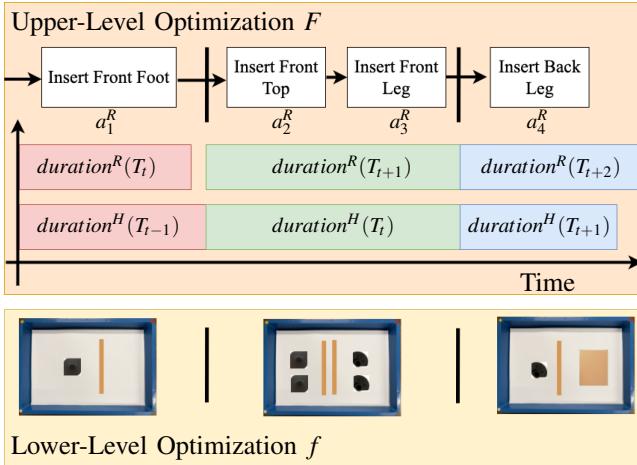


Fig. 3: Overview of the bilevel optimization algorithm for just in time kitting. The upper level optimization segments and schedules tasks, and the lower level optimization arranges parts on the tray. By minimizing the differences between $duration^R(T_{t+1})$ and $duration^H(T_t)$ (shown in green) and the differences between $duration^R(T_{t+2})$ and $duration^H(T_{t+1})$ (shown in blue) as well as other terms in the objective function F (Eq. 3 - 7), the robot determined the task segment T_{t+1} which corresponds to the set of tasks *Insert Front Top* (a_2^R) and *Insert Front Leg* (a_3^R). The task segment T_{t+1} also depended on the kit fitness function f (Eq. 11) which evaluates the kit layout given the parts required P for tasks in T_{t+1} . The robot's role is to prepare parts required for a_2^R and a_3^R in a kit according to the layout determined by f at timestep $t + 1$.

segment. For this, it uses its prior knowledge for how long a given segment will take, both for itself and the human; let us represent this as $duration^H(T_t)$ and $duration^R(T_t)$ respectively. For the task in Figure 2, one possible segmentation is $T = \{T_1, T_2\} = \{\{a_1, a_2\}, \{a_3\}\}$ and a possible schedule is $S = \{0s, 35s\}$. Lastly, the robot must arrange all of the parts for the current segment T_t into the kit — that is, all of the parts required for each task $a \in T_t$.

IV. PROPOSED APPROACH

We formulate our solution as a bilevel optimization, a hierarchical constrained optimization where one problem is embedded in another (i.e., Figure 3). The general formulation of a bilevel optimization problem is:

$$\min_{x \in X, y \in Y} F(x, y) \quad s.t. \quad G(x, y) \leq 0, \quad (1)$$

$$y \in \arg \min_{z \in Z} \{f(x, z) \quad s.t. \quad g(x, z) \leq 0\} \quad (2)$$

where F is the upper-level objective function, G represents the upper-level constraints, f is the lower-level objective function, and g represents the lower-level constraints [4]. In our approach, the upper-level problem Eq. 1 determines the task segmentation and schedule, and the lower-level problem Eq. 2 determines the physical arrangement of parts in the kit.

A. Task Segmentation and Scheduling

We define our notation as follows: $K = \{a_1, a_2, \dots\}$ is an ordered sequence of future tasks to execute (where a_i represents both robot and human tasks a_i^R and a_i^H), i is an index partitioning K to determine the tasks to be included in the next kit, T is a time-ordered sequence of task segments as defined above, $t \in [1, |T|]$ is the current task segment being completed, P is a kit layout defined by a set of part coordinates (part id, x, y) in a kitting tray, G is the task graph encoding precedence constraints for the overall task, d is a scalar constant indicating the time required for a kit delivery, and W_k is a scalar weight hyperparameter for the k th objective function component. We utilize two functions within the objective, *allowed(task, set of completed tasks, task precedence graph)* which returns 1 if *task* can be completed given the *set of completed tasks* and the *task precedence graph* (0 otherwise) and *kit_fitness(kit layout)* which evaluates a *kit layout* for usability (returning a scalar where larger is better).

The optimization for timestep $t + 1$ is defined as follows:

$$\min_{K, i, P} W_1 \sum_{j=1}^{|K|} (1 - \text{allowed}(K_j, \bigcup_{x=1}^t T_x \cup \bigcup_{x=1}^{j-1} K_x, G)) \quad (3)$$

$$- W_2 * i \quad (4)$$

$$+ W_3(d + \sum_{j=1}^i duration^R(K_j) - duration^H(T_t)) \quad (5)$$

$$+ W_4(d + \sum_{j=i+1}^{|K|} duration^R(K_j) - \sum_{j=1}^i duration^H(K_j)) \quad (6)$$

$$- W_5 * \text{kit_fitness}(P) \quad (7)$$

The first term (3) allows us to heavily penalize proposed task sequences K that do not satisfy precedence constraints. The second term (4) can be used to override other terms to prioritize kits covering more tasks. The third term (5) penalizes differences between the human's remaining task time and the duration required to prepare and deliver the proposed next kit. The fourth term (6) is similar, penalizing partitions that are likely to lead to idle times after T_{t+1} . Finally, the fifth term (11) incentivizes kit layouts with high utility, which is solved for by the second level of optimization given a proposed K and i .

After solving for values of K, i, P , we set task segment $T_{t+1} = K_{1:i}$ and increment t . This optimization is run on a greedy basis at each timestep while the robot executes its kitting behaviors. Without loss of generality, i can be defined as a vector of indices that create $|i| + 1$ task segments $T_t = K_{0:[0]}, T_{t+1} = K_{i:[0]:[1]}, \dots, T_{t+|i|} = K_{i:[|i|-1]:|K|}$. Terms (5, 6) would be updated to minimize the differences between human and robot task times for each segment.

The proposed task segmentation and scheduling solution for JIT kitting is an instance of flow-shop scheduling, which has been shown to be NP-hard [6], [8]. We approximate the optimal solution by only considering a limited horizon and partial task plans K of fixed length up to N ($N = 5$ in our experiments).

B. Kit Arrangement

The nested lower-level objective function *kit_fitness* is defined to score the arrangement of parts for a given task segment on the tray used for kit delivery. This task segment T_{t+1} is given by K and i as defined in the previous section.

Let P denote the set of parts required for the tasks in T_{t+1} and their coordinates on the 2D plane of the kitting tray. Let $\mathbf{P}_{j,x}$ and $\mathbf{P}_{j,y}$ denote the x and y coordinates of the j th part's centroid in the kit. These coordinates are solved for with the lower optimization, with an objective function that maximizes kit fitness. We first define terms of the *kit_fitness* objective to prioritize logical grouping of items on the tray:

$$D_{diff}(P) = \sum_{k=1}^{|P|} \sum_{\substack{j=k+1: \\ Q(P_j) \neq Q(P_k)}}^{|P|} \sqrt{(P_{k,x} - P_{j,x})^2 + (P_{k,y} - P_{j,y})^2} \quad (8)$$

$$D_{same}(P) = \sum_{k=1}^{|P|} \sum_{\substack{j=k+1: \\ Q(P_j) = Q(P_k)}}^{|P|} \sqrt{(P_{k,x} - P_{j,x})^2 + (P_{k,y} - P_{j,y})^2} \quad (9)$$

where $Q(p)$ denotes the *part type* of part p , such as "M6 screw" or "connector." Equation 8 sums the Euclidean distance between all parts of different types, while 9 does the same for all parts of the same type.

We also define a term to penalize overlap between items on a tray, as we wish to discourage the robot from piling objects on top of one another. Let $BB(p, x, y)$ denote the *bounding box* of part p centered at point (x, y) . We define Z as the total overlapping area between the bounding boxes of all parts:

$$Z(P) = \sum_{k=1}^{|P|} \sum_{j=k+1}^{|P|} BB(P_k, P_{k,x}, P_{k,y}) \cap BB(P_j, P_{j,x}, P_{j,y}) \quad (10)$$

Lastly, we use the terms described above to define our kit fitness objective function, with terms K, i defined by the upper-level optimization and W_6 a scalar hyperparameter to modulate the optimization's focus on discouraging part overlap:

$$\min_P D_{same}(P) - D_{diff}(P) - W_6 * Z(P) \quad (11)$$

$$\begin{aligned} \text{subject to: } & \bigcup_{k=1}^{|P|} BB(P_k, P_{k,x}, P_{k,y}) \subset \text{area}_{kit} \\ & \bigcup_{j=1}^i \text{parts_in}(K_j) \in P \end{aligned}$$

The objective function maximizes distances of parts with different types, minimizes distances of parts with the same type, and penalizes overlaps. The constraint ensures that the 2D space taken up by parts in the kit does not exceed or extend beyond the tray area and that all necessary parts are included in the kit.

We use the cross entropy (CE) method to solve the kit arrangement problem. CE is a stochastic algorithm that

provides an adaptive procedure for solving combinatorial optimization problems and has asymptotic convergence properties [5]. Other stochastic methods such as simulated annealing [19] and genetic algorithms [10] can also be used.

Algorithm 1 Cross Entropy Method for kit arrangement

```

Require: list of parts  $P$ , sample count to keep  $c$ 
    initialize  $\mu(P_{k,x}, P_{k,y}, \theta_i \forall k \in 1 : |P|), \Sigma$  randomly
    while not converged and max iterations not reached do
         $samples \sim \mathcal{N}(\mu, \Sigma)$ 
         $scores \leftarrow sorted(score(samples))$        $\triangleright$  score samples
        and sort from largest to smallest
         $\mu \leftarrow mean(samples[:c])$ 
         $\Sigma \leftarrow covariance(samples[:c])$   $\triangleright$  recompute mean and
        covariance from new samples
    end while
    return  $\mu$ 

```

Algorithm 1 shows the implementation for CE. We first randomly initialize the mean vector and covariance matrix. The mean vector contains the position (x and y) and orientation (θ) of the parts that have to be placed on the tray. While the solution hasn't converged or the maximum number of iterations (100) has not been reached, we draw samples from a Gaussian distribution (200 in our experiments) with the current estimated mean and covariance. The samples are scored using Equation 11 and sorted from largest to smallest. Then we use the top c samples to recompute the mean and covariance ($c = 30$ in our experiments). The algorithm returns the estimated mean after convergence or after the maximum number of iterations.

V. EXPERIMENTAL VALIDATION

With the goal of evaluating our proposed system under a variety of logistic delay scenarios, we first perform a user study to collect user experience data. We then use this study as a basis to perform a series of simulated shop floor evaluations of the robot's kitting delivery performance under multiple conditions varying supply failure likelihood and delivery distance between kitting and assembly stations.

A. User Study

We had 19 people (10 males, 9 females) participate in our IRB-approved study, ranging in age from 22 to 31 ($M = 25.47$, $SD = 2.37$). One data point was discarded because the Sawyer robot lost connection to the study computer in the middle of the experiment. Participants were tasked with the construction of a flat-pack furniture table [21] shown in Figure 4. Table assembly involves connecting four feet, four legs, four connectors, and a flat surface plank by snapping the pieces together and securing with screws and nuts.

1) Experimental Design: We use a within-subjects experiment to evaluate our optimization approach and two baseline conditions: (a) *Whole Assembly* is a baseline which resembles traditional kitting strategies where all the required parts for a single unit are placed on the kitting tray prior to

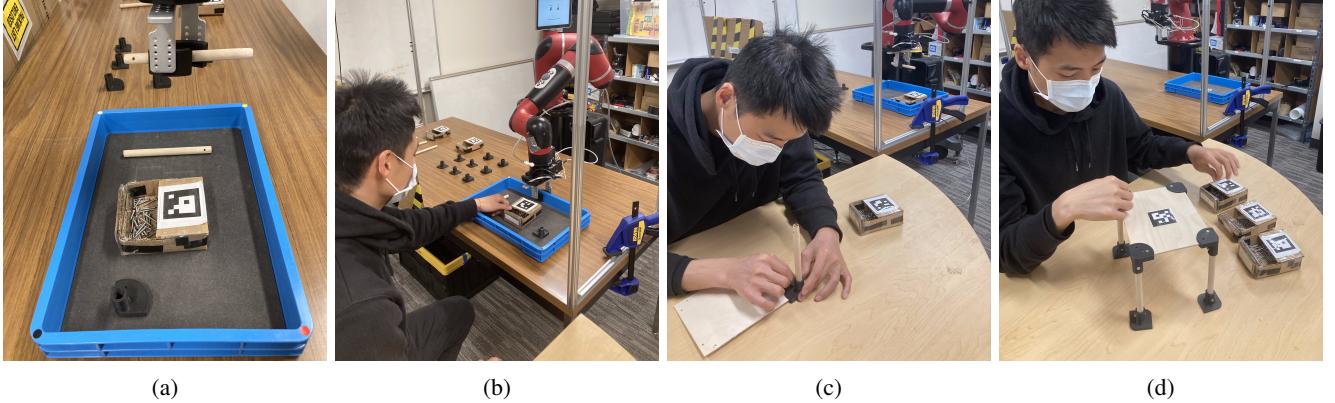


Fig. 4: (a) The robot arranges parts on the tray according to the lower-level optimization. (b) The robot delivers a finished kit to the user. (c) The user begins the assembly task while the robot prepares the kit for the next tasks. (d) The user finishes assembling a flat-pack furniture table.

assembly, (b) *Single Task* is a human-designed JIT schedule where the robot delivers parts for a single task at a time as segmented by the task graph itself (one kit per vertex), and (c) *Optimized* is the result of applying the proposed bilevel optimization algorithm to the assembly task. To minimize learning effects, we counterbalance the condition orderings and ask the participants to practice assembling part of the table before the experiment.

The assembly task is divided into a total of 12 tasks, i.e. 3 tasks per leg of the table. The precedence constraints (see Section III) dictate that the feet and connecting joint have to be secured to the leg before the leg can be secured to the plank. The robot’s tasks include delivering the screws, nuts, four legs, four feet, and four joints to the human participant via a kitting tray. The human participant and the robot have separate workspaces, and the human participant is instructed to not enter the robot’s workspace during the experiment. Figure 1 shows the experiment setup.

2) *Metrics*: The objective metrics used to assess the efficiency of our proposed approach are the total task time and the human idle time. The *total task time* is the total experiment time for assembling a single table. The *human idle time* is defined as the duration of the experiment in which the participant was not actively assembling parts. A post-experiment survey was used to assess the user experience benefits of our proposed approach: participants were asked to rank each trial based on the robot’s Usefulness, Intuitiveness, and Efficiency during that trial.

3) *Hypotheses*:

- **H1:** The *Optimized* condition will have a makespan lower-bounding *Single Task* and *Whole Assembly* conditions with reduced idle time that is proportional to delivery duration.
- **H2:** Participants will perceive the *Optimized* condition more highly along subjective measures (usefulness, intuitiveness, and efficiency) than the *Single Task* and *Whole Assembly* conditions.

4) *Study Protocol*: After a brief introduction to the robot’s role in part delivery, the human participant was presented

with a printed guide document, which included step-by-step instructions in both written and visual form. The participant was asked to practice constructing a single leg of the table before the experiment began.

During the experiment, the required parts were initially placed in the robot workspace. The robot kitted the assembly parts by placing some number of them onto a tray, and then pushing the tray towards the human. The participant was notified to retrieve the parts on the tray after the robot did this. The same assembly was repeated for all three conditions. The robot returned to its kitting task once the participant

5) *Implementation Details*: We implemented the optimization framework on a Sawyer Research Robot, using the Robot Operating System (ROS) [13]. We installed an Intel RealSense D435 Depth Camera above the robot workspace and used the Facebook Detectron2 algorithm [20] for image segmentation of the assembly parts. We used the ArUco library [9] to detect markers placed on the screw boxes and tray to read their pose.

B. Shop Floor Simulation

While the user study provides useful insights for user preferences and task efficiency for a single assembly, we use discrete event simulation (DES) to evaluate our proposed approach when multiple assemblies of the furniture table are performed in sequence. We evaluate the framework under various assembly part arrival time distributions and part-feeding machine breakdown conditions (i.e. the part is not available to the robot until after the machine is repaired). DES is able to explicitly model events using probability distributions and answers questions such as the line throughput and utilization [16].

1) *Experimental Design*: We simulated the assembly of ten flat-pack furniture tables. For each assembly, we randomly sampled a participant’s task times and the robot’s task times as recorded from the user study detailed in V-A. The arrival time of assembly parts to the robot’s workspace is modeled by an exponential distribution with Mean Arrival

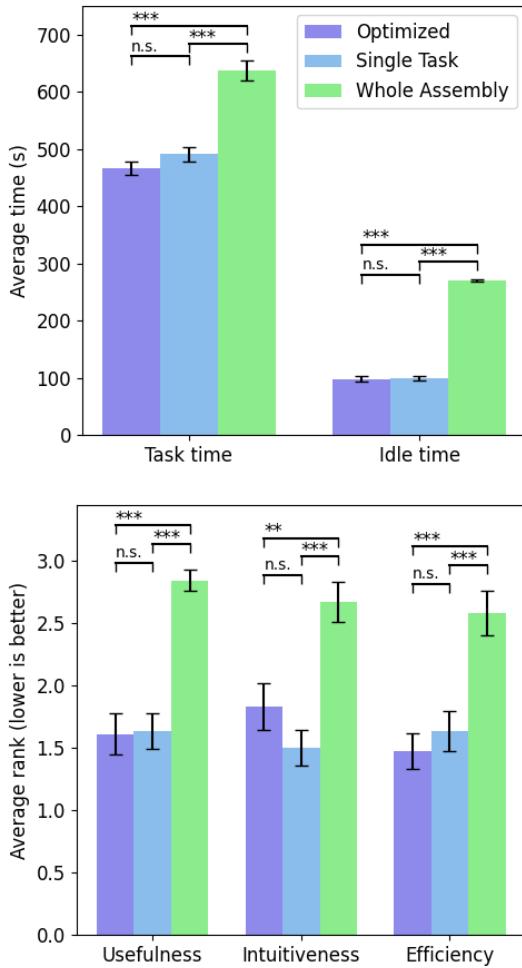


Fig. 5: **Top:** Total task time and human idle time across all three conditions. Both the *Optimized* and *Single Task* approaches had significantly lower total times in both metrics than in the *Whole Assembly* condition. **Bottom:** Total user ranking scores for each condition across three subjective metrics: usefulness, intuitiveness, and efficiency. The *Whole Assembly* condition had significantly worse scores than the algorithmic conditions across all three rankings. The *Single Task* and *Optimized* conditions did not vary significantly on any of the above metrics, likely due to the proximity of the robot to the human (no delivery delays).

** indicates $p < .001$; *** indicates $p < .0001$.

Time denoted by MAT . In other words, the arrival of parts is a Poisson process with rate $1/MAT$. Since increasing the arrival times of all the parts linearly increases the total task times, we only vary the MAT of the leg and foot part types. We also model the machine breakdown for the feeding of the leg and foot parts with an exponential distribution. The mean time to failure of the machines is denoted by $MTTF$. Each machine has a fixed repair time of 30 seconds, after which it starts feeding parts again according to the part arrival process with rate $1/MAT$.

2) Hypothesis:

- **H3:** When simulating multiple assemblies of the task, the *Optimized* condition will significantly outperform baseline kitting strategies in both total task time and human idle time in the presence of logistical delays.

3) *Implementation Details:* The simulation is implemented in Python using the SimPy library. In order to account for assembly part shortages due to slow arrival times or machine breakdowns, we add an additional term to the upper-level objective function:

$$W_7 \sum_{k=1}^{|P|} U(P_k) \quad (12)$$

where $U(p)$ returns 1 if part p is currently unavailable for the robot to place on the tray and 0 otherwise.

VI. RESULTS

A. User Study

1) *Objective measures:* Total task time and human idle time measures were analyzed using the one-way analysis of variances (ANOVA) with experimental condition as an independent variable. Post-hoc analysis used Tukey's HSD test for multiple comparisons to test for effects between condition pairs. The results are summarized in Figure 5.

Total task time: The effect of condition significantly influenced total task time, $F(2, 34) = 57.60, p < .0001$, with a significant reduction in task time between the *Whole Assembly* condition and the *Single Task* condition ($p < .0001$, 95% C.I. = [-188.211, -103.851]), as well as between the *Whole Assembly* and *Optimized* conditions ($p < .0001$, 95% C.I. = [-213.213, -128.853]).

Human idle time: The effect of experimental condition significantly influenced human idle time, $F(2, 34) = 709.87, p < .0001$; there was a significant reduction in idle time between the *Whole Assembly* and *Single Task* conditions ($p < .0001$, 95% C.I. = [-184.465, -158.586]), as well as between the *Whole Assembly* and *Optimized* conditions ($p < .0001$, 95% C.I. = [-186.020, -160.141]).

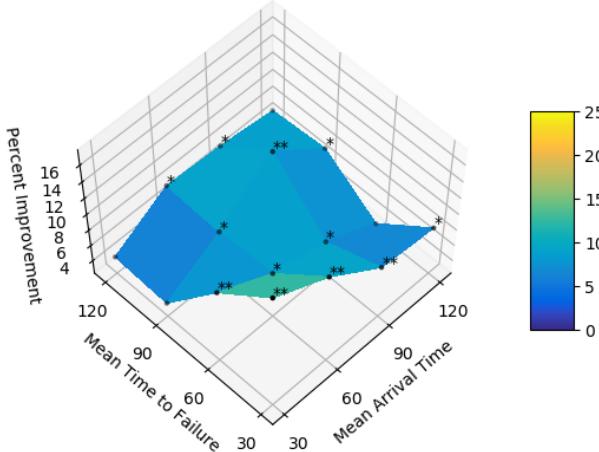
2) *Subjective measures:* We analyzed rank scale data on our post-experiment survey using a nonparametric Kruskal-Wallis Test with experimental condition as a fixed effect. Post-hoc comparisons used the Wilcoxon method for analyzing rank significance between condition pairs. The results are summarized in Figure 5.

B. Simulation

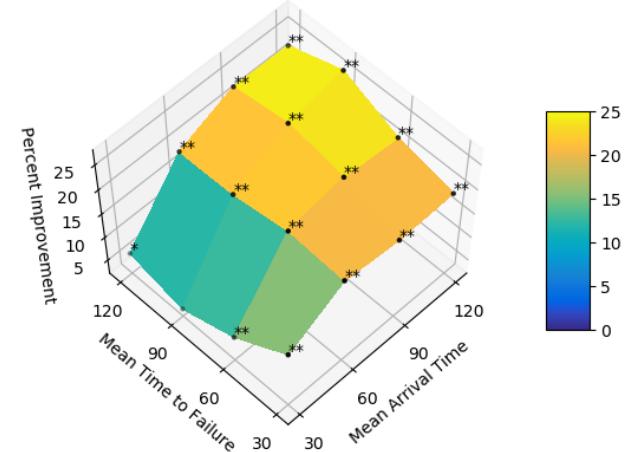
In simulation, we compare the task time and human idle time outcomes between *fixed* strategies (*Single Task* and *Whole Assembly*) and the *Optimized* strategy. Percent improvements in task time and human idle time between conditions are summarized in Figure 6. Results were analyzed using the same methods as in Section VI-A.

VII. DISCUSSION

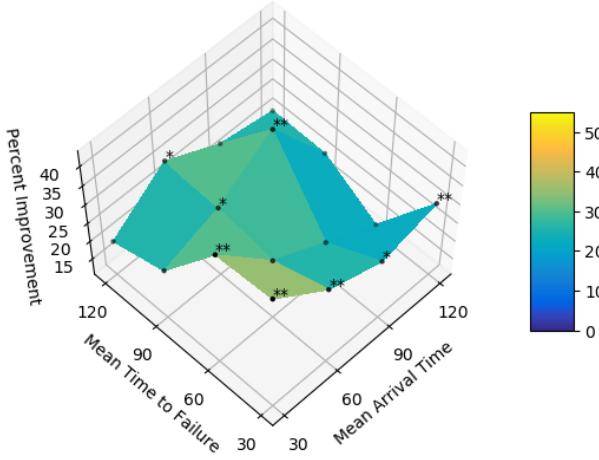
Hypothesis **H1** regarding task time and idle time is supported by the user study results. We suspect that *Optimized* is similar to *Single Task* due to the high variance in the



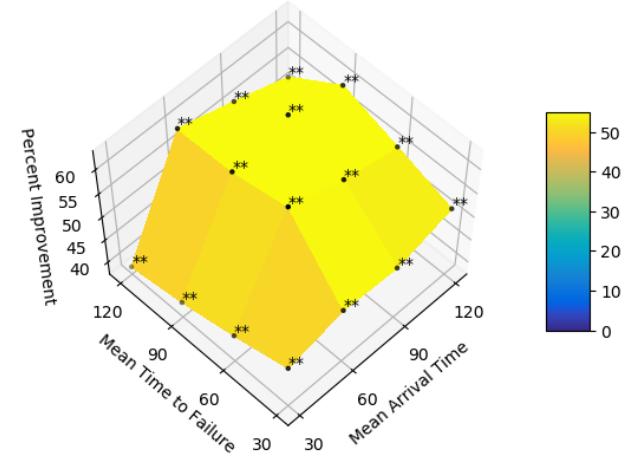
(a) Percent improvement in total task time of *Optimized* over *Single Task*.



(b) Percent improvement in total task time of *Optimized* over *Whole Assembly*.



(c) Percent improvement in human idle time of *Optimized* over *Single Task*.



(d) Percent improvement in human idle time of *Optimized* over *Whole Assembly*.

Fig. 6: Comparisons of Optimized approach with other approaches in simulation. (a) and (b) show the percent improvement in total task time of the *Optimized* approach over the *Single Task* and *Whole Assembly* conditions respectively. With low part shortage (i.e. low MAT), *Optimized* is most advantageous over *Single Task* because *Single Task* has many more tray deliveries which increase the total robot task time. With high part shortage (i.e. high MAT), *Optimized* is most advantageous over *Whole Assembly* because *Optimized* adapts the kitting strategy according to which parts are available. (c) and (d) show the percent improvement in human idle time of the *Optimized* approach over the *Single Task* and *Whole Assembly* conditions respectively. The trend for human idle time is similar to that of total task time. * denotes $p < .05$ while ** denotes $p < .001$.

participants' task times and the absence of delivery distance (the human and robot were co-located across a table). The average human task time (i.e. non-idle time) for a single assembly and across all conditions was 374.78 seconds with a standard deviation of 61.14 seconds.

Hypothesis **H2** is supported; the *Optimized* condition had significantly lower ranking scores (i.e. were ranked better) in usefulness, intuitiveness, and efficiency than *Whole Assembly*. However, there was no significant difference in rankings of *Single Task* and *Optimized* on any of the three subjective metrics (all $p \geq .05$). From the post-experiment interviews, participants found the *Optimized* or *Single Task*

conditions most efficient because "it gave me the parts in timely manner", and "[it has] less waiting time." This suggests that participants preferred just-in-time kitting strategies. Some participants also indicated they found the just-in-time approaches less cognitively demanding; "[The robot is] guiding me through the most efficient way to complete the assembly job ... the robot lifted part of my pressure to plan it."

Hypothesis **H3** is supported; the total task time and human idle time are significantly shorter for *Optimized* than either *Single Task* or *Whole Assembly* for most of the scenarios with logistic delays. *Optimized* is most advantageous over

Whole Assembly when there is a high part shortage (high MAT) because *Whole Assembly* waits for all the parts to be kitted before delivering the kit to the assembly station. When a part is delayed, *Whole Assembly* waits until the part becomes available. While *Single Task* also suffers from logistic delays, the effect is less prominent because kits are delivered with partial assembly parts. The *Optimized* approach is able to dynamically determine which tasks to deliver parts for in order to minimize idle times and maximize efficiency.

VIII. CONCLUSIONS

This work introduces a bilevel optimization approach for robot kitting and demonstrates its ability to reduce both overall task time and human and robot idle times for a furniture assembly task. In a user study, we evaluated this approach against a generic whole-kit assembly (*Whole Assembly*) and a human-designed just in time approach (*Single Task*), and found that just in time kitting had quicker task completion and were rated more highly by users on a number of subjective metrics. Simulating longer and more varied task environments revealed that the online optimized approach demonstrates significant performance improvement over the fixed kitting strategies in more realistic settings that include logistic delays.

Moreover, the user study suggested two additional avenues of future research: online estimates of human task time to better minimize idle times and the addition of human factors into the optimization framework. Lastly, we plan to investigate which factors best improve the efficiency and intuitiveness of kitting tray designs.

REFERENCES

- [1] M. E. A. Boudella, E. Sahin, and Y. Dallery, “Kitting optimisation in just-in-time mixed-model assembly lines: assigning parts to pickers in a hybrid robot-operator kitting system,” *International Journal of Production Research*, vol. 56, no. 16, pp. 5475–5494, 2018.
- [2] Y. A. Bozer and L. F. McGinnis, “Kitting versus line stocking: A conceptual framework and a descriptive model,” *International Journal of Production Economics*, vol. 28, no. 1, pp. 1–19, 1992.
- [3] A. C. Caputo, P. M. Pelagagge, and P. Salini, “A model for planning and economic comparison of manual and automated kitting systems,” *International Journal of Production Research*, vol. 59, no. 3, pp. 885–908, 2021.
- [4] B. Colson, P. Marcotte, and G. Savard, “An overview of bilevel optimization,” *Annals of Operations Research*, vol. 153, no. 1, pp. 235–256, Sep 2007. [Online]. Available: <https://doi.org/10.1007/s10479-007-0176-2>
- [5] P.-T. De Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein, “A tutorial on the cross-entropy method,” *Annals of operations research*, vol. 134, no. 1, pp. 19–67, 2005.
- [6] K. Ewacha, I. Rival, and G. Steiner, “Permutation schedules for flow shops with precedence constraints,” *Operations research*, vol. 38, no. 6, pp. 1135–1139, 1990.
- [7] A. Fansuri, A. Rose, N. N. Mohamed, and H. Ahmad, “The challenges of lean manufacturing implementation in kitting assembly,” in *IOP Conference Series: Materials Science and Engineering*, vol. 257, no. 1. IOP Publishing, 2017, p. 012069.
- [8] M. R. Garey, D. S. Johnson, and R. Sethi, “The complexity of flowshop and jobshop scheduling,” *Mathematics of operations research*, vol. 1, no. 2, pp. 117–129, 1976.
- [9] S. Garrido-Jurado, R. Muñoz-Salinas, F. Madrid-Cuevas, and R. Medina-Carnicer, “Generation of fiducial marker dictionaries using mixed integer linear programming,” *Pattern Recognition*, vol. 51, 10 2015.
- [10] D. E. Goldberg and J. H. Holland, “Genetic algorithms and machine learning,” *Machine Learning*, 1988.
- [11] L. Jeanson, J. C. Bastien, A. Morais, and J. Barcenilla, “Tell me how you kit, i’ll tell you how you think,” in *H-Workload 2018: 2nd International Symposium on Human Mental Workload: Models and Applications*, p. 232.
- [12] J. Joudni, P. Conradi, J. Van Den Bergh, and J. Saldien, “Understanding and exploring operator needs in mixed model assembly,” in *EISMS19, Workshop on Research and Practice Challenges for Engineering Interactive Systems while Integrating Multiple Stakeholders Viewpoints*, 2019.
- [13] A. Koubâa et al., *Robot Operating System (ROS)*. Springer, 2017, vol. 1.
- [14] R. Maderna, M. Poggiali, A. M. Zanchettin, and P. Rocco, “An online scheduling algorithm for human-robot collaborative kitting,” in *2020 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2020, pp. 11430–11435.
- [15] M. S. Nikam, A. Joel, and J. Dutta, “Design and implementation of kitting trolley for just in time production in textile industry,” *International Research Journal of Engineering and Technology (IRJET)*, vol. 5, no. 4, pp. 1301–1304, 2018.
- [16] N. Prajapat and A. Tiwari, “A review of assembly optimisation applications using discrete event simulation,” *International Journal of Computer Integrated Manufacturing*, vol. 30, no. 2-3, pp. 215–228, 2017.
- [17] S. Shaikh, S. Cobb, D. Golightly, J. Segal, and C. Haslegrave, “Effects of takt time on physical and cognitive demands in a mixed model assembly line and a single model assembly line,” in *Contemporary Ergonomics and Human Factors 2012: Proceedings of the international conference on Ergonomics & Human Factors 2012, Blackpool, UK, 16-19 April 2012*. CRC Press, 2012, p. 137.
- [18] B. Townsend, *Line Balancing And Jit Kitting*. Boca Raton, Fl: Crc Press, 2012.
- [19] P. J. Van Laarhoven and E. H. Aarts, “Simulated annealing,” in *Simulated annealing: Theory and applications*. Springer, 1987, pp. 7–15.
- [20] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, “Detectron2,” <https://github.com/facebookresearch/detectron2>, 2019.
- [21] S. Zeylikman, S. Widder, A. Roncone, O. Mangin, and B. Scassellati, “The hrc model set for human-robot collaboration research,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1845–1852.
- [22] B. Zhou and Z. He, “A static semi-kitting strategy system of jit material distribution scheduling for mixed-flow assembly lines,” *Expert Systems with Applications*, vol. 184, p. 115523, 2021.
- [23] X. Zhu, S. J. Hu, Y. Koren, and S. P. Marin, “Modeling of Manufacturing Complexity in Mixed-Model Assembly Lines,” *Journal of Manufacturing Science and Engineering*, vol. 130, no. 5, 08 2008, 051013. [Online]. Available: <https://doi.org/10.1115/1.2953076>