Risk-Aware Planning of Collaborative Mobile Robot Applications with Uncertain Task Durations

Anders Lager[1,2], Branko Miloradović[2], Giacomo Spampinato[1], Thomas Nolte[2], Alessandro V. Papadopoulos[2]

***Abstract*— The efficiency of collaborative mobile robot applications is influenced by the inherent uncertainty introduced by humans' presence and active participation. This uncertainty stems from the dynamic nature of the working environment, various external factors, and human performance variability. The observed makespan of an executed plan will deviate from any deterministic estimate. This raises questions about whether a calculated plan is optimal given uncertainties, potentially risking failure to complete the plan within the estimated timeframe. This research addresses a collaborative task planning problem for a mobile robot serving multiple humans through tasks such as providing parts and fetching assemblies. To account for uncertainties in the durations needed for a single robot and multiple humans to perform different tasks, a probabilistic modeling approach is employed, treating task durations as random variables. The developed task planning algorithm considers the modeled uncertainties while searching for the most efficient plans. The outcome is a set of the best plans, where no plan is better than the other in terms of stochastic dominance. Our proposed methodology offers a systematic framework for making informed decisions regarding selecting a plan from this set, considering the desired risk level specific to the given operational context.**

I. Introduction

Integrating collaborative robot applications in the industrial landscape started over a decade ago [13]. While robotic utilization enhances productivity and ergonomic conditions by managing assistive, repetitive, and strenuous tasks, the contemporary industrial trend towards mass customization values the unique skills, adaptability, dexterity, and problem-solving capabilities of human workers [1]. Our paper specifically focuses on mobile robots in collaborative applications, appreciating their flexibility to execute diverse tasks across various locations, catering to the needs of human workers [21]. In collaborative industrial environments characterized by semi-structured and dynamic settings, the duration of robot routing can be influenced by temporary obstacles and concurrent human and robot activities. Additionally, uncertainties in robot task durations arise from unpredictable placements of required parts or tools and the possibility of task failures, leading to retries. Similarly, the duration of human tasks introduces uncertainty, influenced by variables such as workload, fatigue, availability, and location.

This paper aims to study the uncertainty associated with the duration of tasks performed by robots and humans, affecting collaborative plans' accuracy. Traditional deterministic planning methods fail to account for potential deviations from the estimated duration of tasks. Additionally, identifying an optimal plan is difficult due to the inherent uncertainty in task durations and the subjectivity involved in choosing the "best" plan, which can impact the risk tolerance of a human planner. For example, a plan with a given probability of completing the tasks within a certain time limit may be preferred over a plan with a guaranteed upper bound for the makespan. While the former may result in a lower makespan, it can sometimes lead to a higher makespan than the upper bound of the latter. Taking a medium risk may reduce the expected makespan over multiple runs. Taking a larger risk can be motivated if a low makespan gives a reward, whereas the distinction between a shorter or longer additional delay might not be crucial. This paper investigates industrial task planning problems for a mobile robot collaborating with humans in dynamic environments. To model the planning problem, we use Robot Task Scheduling Graph (RTSG) [8]. This is motivated by the intuitiveness of the representation rather than its expressiveness, benefiting domain experts who have broad knowledge of the application but no in-depth knowledge of all system parts, e.g., robot programming. This paper extends the RTSG model to enhance the flexibility of work descriptions by considering concurrent human tasks and robot tasks with inter-dependencies. The stochastic modeling approach presented in this paper introduces uncertainty into task and routing durations using random variables with unrestricted probability distributions. When concrete data is lacking, initial simplifications of input distributions, such as uniform distributions, are employed. These distributions can be iteratively refined through data collection during system execution, progressively improving accuracy. The main contribution is a novel task planning methodology involving computing optimized plans where uncertainties are considered during plan generation. The output is a set containing the best plans, where no plan is better than the other in terms of stochastic dominance. Notably, the methodology allows a human planner to make an informed decision on the most suitable plan from this candidate set by providing a risk level value and/or by visually inspecting probabilistic makespan distributions of the candidates. This approach is inspired by Mixed-Initiative Planning [5]. The novel task planning methodology leverages a Branch-and-Bound (B&B) algorithm [9] able to solve planning problems modelled with RTSG, extended to handle stochastic durations. It explores alternative sub-sequences, with their durations derived as probability density functions. One contribution is the derivation of unrestricted probability

This work is funded by The Knowledge Foundation (KKS), projects ARRAY and MARC, by the Swedish Research Council (VR), project PSI.

[1]ABB AB, Västerås, Sweden. [2]Division of Intelligent Future Technologies, Mälardalen University, Västerås, Sweden. Email: {name.surname}@se.abb.com; {name.surname}@mdu.se

distributions to represent makespans of collaborative plans. Such a distribution is a realistic estimate of the makespan range that may occur if the plan is executed. Another contribution is a novel pruning strategy, which uses the first-order stochastic dominance property to ensure safe pruning. It guarantees that pruning never eliminates a superior plan in a stochastic sense. Additionally, we contribute by proposing *stochastic set dominance* as a criterion to filter full-length plans into a candidate set containing only the very best plans, where no plan's makespan dominates others in a stochastic sense. To validate the approach, it is benchmarked against a deterministic counterpart. Additionally, Monte Carlo simulations [17] are conducted to verify the correctness of stochastic makespan computations generated by the planning algorithm. These evaluations demonstrate the ability of our methodology to provide more efficient plans and to support risk-aware task planning under uncertain conditions.

II. Related work

In a recent approach for task planning in collaborative assembly applications, a policy for task allocation based on the current state is trained to optimize future rewards [11]. This work and all other related works differs from ours, by not providing a set of plans optimized for different risk levels. Our approach accounts for uncertainties in the planning problem, while some other approaches solve a deterministic planning problem and handle uncertainties during execution [6] [10], potentially providing a suboptimal plan. Probabilistic Simple Temporal Networks (PSTN) are used to model scheduling problems with temporal constraints using random variables to represent uncertain task durations [18]. In general, PSTN addresses *scheduling risks* by searching for robust plans to minimize or control the risk for plan execution failures caused by violation of temporal constraints, e.g., missed deadlines. The problem addressed in this paper has no temporal constraints, and the type of risk addressed is different, i.e., the risk of getting a longer duration than expected at plan execution. In some works, the primary focus of a robot is assisting by anticipating the next human task and providing needed tools or parts just in time [7], [22]. In our work, the robot serves multiple humans, and a long-sighted sequence of robot actions is planned. One scheduling approach used Monte Carlo simulations in a receding horizons approach to estimate the cost distributions of alternative execution sequences of robot tasks and human tasks with uncertain durations [2]. A receding horizons approach naturally limits the growth of a search tree to a manageable level. For our problem type, where the objective is to minimize the makespan, a plan may become greedy and less optimal with a short-sighted look ahead, causing later costs to dominate. Additionally, we compute cost distributions in a closed form, giving a qualitative advantage when comparing alternative plans and sub-sequences, including the possibility of safe pruning decisions to reduce the growth of a search tree. In a receding horizon scenario similar to [2], the uncertainties in durations of human and robot tasks were modeled and propagated for alternative task sequences as triangular fuzzy sets, which in an actual application provided a better plan optimization than an approach where task durations were modeled more simplistic, as uniform distributions [3]. This result motivates the usage of richer representations, as proposed in this paper, to represent uncertain durations when modeling collaborative applications. Similar to our approach, [16] models routing and action durations as random variables and uses a framework of stochastic operations to compute stochastic start and completion times of tasks for different scenarios where multiple robots share a mutually exclusive resource. However, probability distributions are limited to be Gaussian, while our approach does not impose such restrictions. In [15], this Gaussian framework was applied to a task planning problem of a replenishment agent serving other agents, where a finite-horizon schedule of tasks is computed with a B&B approach, including pruning of branches where a conservative estimate of the minimum cost is higher than currently the best solution. We propose a novel pruning strategy based on stochastic dominance, proven to be safe for this application when comparing *unrestricted* distributions of search nodes representing the same state. A proactive scheduling approach for a Job shop problem was proposed by [12], using durations modeled as random variables. As a part of the solution, the sequence of operations on a machine was computed with a B&B algorithm, minimizing a weighted combination of expectation and variance of the completion time. A safe pruning criterion was proposed, using Stably Stochastic Dominance, providing an ordering of alternative sequences based on expectation and variance of operations. While this approach is motivated by the need to find a robust plan with limited time variations, our approach computes a candidate set of plans, whose makespans are stochastically dominated by alternative plans, thereby providing the most *efficient* choices for any risk level.

III. Modeling the planning problem

In this section, we define and model the planning problem. We present the modeling of task durations as random variables and provide related definitions for later reference.

A. Problem description and assumptions

A mobile industrial robot is used to deliver parts to different assembly workstations. At these stations, sub-assemblies are made either by human workers or by the robot itself. The robot fetches the finished sub-assemblies, and finally, all sub-assemblies are delivered to another station for further processing. It is assumed there is no load restriction related to the mobile robot's ability to carry parts and sub-assemblies. Task allocation is static, i.e., there are robot tasks and human tasks. While considering the dependencies between robot tasks and human tasks, we assume human tasks are mutually independent. There is no physical interaction between the robot and humans and their interaction level can be categorized as synchronized, i.e., they share the same space at the delivery and fetching locations, but not at the same time [14]. The effects of tasks are assumed to be deterministic, i.e., all tasks will eventually succeed. However, task and routing durations are uncertain. The goal is to compute an offline plan [4] that minimizes the uncertain makespan while considering the aversion or willingness to risk. In this context, the risk is related to efficiency. Increasing the risk means we increase the probability of reaching shorter makespans while also accepting the risk of sometimes reaching longer makespans than before. The risk willingness may influence what plan is considered to be the best.

B. Modeling a collaborative planning problem

The planning problem is modeled with an RTSG model, exemplified in Fig. 1. It is a directed acyclic graph giving an intuitive workflow description of how tasks can be sequenced. The *Start* node and the *Goal* node represent the initial state and the desired goal state, respectively. Task nodes have rectangular shapes. Edges (or paths) represent precedence constraints, e.g., $DL1$ must precede $MV1$ while $DL1$ and $DL2$ may be performed in any order. Alternative branches are modeled between OR-Fork (||F) and OR-Join (||J) node pairs, and in this model, they describe the alternative selections of human assembly ($HA1$) or robot assembly ($RA1$). Lock node pairs (+L,-L) indicate a branch section where robot tasks must be scheduled in an uninterrupted sequence. Here, the robot movement to a human assembly station ($MV1$) must be followed by the robot picking ($PI1$) the sub-assembly provided by the human ($HA1$). AND-Fork nodes (&F) create parallel branches while AND-Join nodes (&J) join branches. In this work, the RTSG modeling formalism has been extended to represent *human tasks*, i.e., tasks allocated to humans (e.g., $HA1$). Human tasks may be performed concurrently with robot tasks, and they need to follow the scheduling constraints set by the RTSG model. Additionally, the need to synchronize human tasks with robot tasks is addressed. A new node type, AND-JoinSync (&JS), has multiple incoming branches and a single outgoing branch. The &JS node blocks the robot from proceeding with succeeding robot tasks until all preceding human tasks are completed, potentially causing some robot wait time. For example, after moving to an assembly station ($MV1$), the robot may not pick the sub-assembly ($PI1$) until the human assembly task ($HA1$) has provided the assembly.

C. Preliminaries and definitions

Durations for performing robot tasks and human tasks are indexed variables. The routing duration $R_{\tau,\tau'}$ between robot tasks, $\tau, \tau' \in T$ where T is the set of all robot tasks. The duration A_τ to perform a robot task $\tau \in T$ and the duration B_h to perform a human task $h \in H$, where H is the set of all human tasks. This work models these durations (B_h, A_τ, $R_{\tau,\tau'}$) as independent random variables without assuming any specific probabilistic distributions.

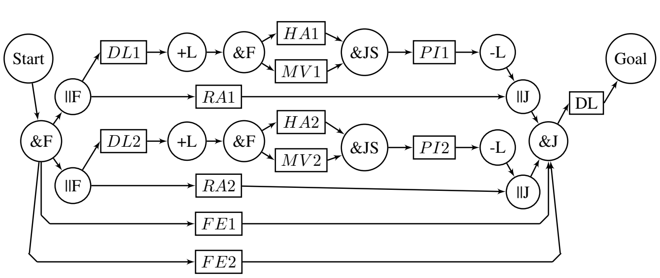

Fig. 1: RTSG model with 2 human assembly tasks.

Definition 1 (Random Variable)**.** *A random variable X on the probability space $(\Omega, \mathcal{F}, \mathbb{P})$ is a measurable function $X : \Omega \to \mathbb{R}$ such that $\{\omega \in \Omega : X(\omega) = x\} \in \mathcal{F}$ for all $x \in \mathbb{R}$.*

Definition 2 (Expected value)**.** *Given a random variable X, its* expected value $\mathbb{E}[X] \in \overline{\mathbb{R}}$ *is a measure of the central tendency or average value of the possible outcomes of X:*

$$\mathbb{E}[X] \triangleq \int_{\omega\in\Omega} X(\omega)\, d\mathbb{P}.$$

Definition 3 (Variance)**.** *Given a random variable X, its* variance*, denoted by $\mathbb{V}[X]$, is a measure of the dispersion or spread of the possible outcomes of X.*

$$\mathbb{V}[X] \triangleq \mathbb{E}\Big[\big(X - \mathbb{E}[X]\big)^2\Big]$$

Definition 4 (Standard deviation)**.** *The standard deviation $\sigma[X]$ of a random variable X is the square root of its variance:*

$$\sigma[X] \triangleq \sqrt{\mathbb{V}[X]}.$$

Definition 5 (Probability density function (PDF))**.** *The probability density function $f_X(x)$ of a random variable X is defined as:*

$$f_X(x) = \mathbb{P}[\omega \in \Omega \mid X(\omega) = x]$$

Definition 6 (Cumulative distribution function (CDF))**.** *The cumulative probability distribution function $F_X(x)$ of a random variable X is defined as:*

$$F_X(x) = \mathbb{P}[\omega \in \Omega \mid X(\omega) \leq x]$$

Definition 7 (Percentile)**.** *The k-th percentile of a probabilistic distribution $f_X(x)$ is defined as:*

$$p_k = \inf\{x : F_X(x) \geq k\}, \quad 0 < k < 1.$$

Definition 8 (First-Order Stochastic Dominance [19])**.** *Consider two random variables, X and Y, with CDFs F_X and F_Y. X has a first-order stochastic dominance over Y, if and only if $\forall x$, $F_X(x) \leq F_Y(x)$, and $\exists x$, $F_X(x) < F_Y(x)$. The stochastic dominance is denoted in the following as $X \succeq Y$. If the given condition is not fulfilled, this is denoted $X \not\succeq Y$.*

Definition 9 (Stochastic Set Dominance)**.** *Consider one random variable X and a set of random variables $S = \{Y_1, \ldots, Y_a\}$, with CDFs F_X and $F_{Y_1}, \ldots, F_{y_a}$. X has a set dominance over S, if and only if $\forall x$, $F_X(x) \leq \max\{F_{Y_1}(x), \ldots, F_{Y_a}(x)\}$, and $\exists x$, $F_X(x) < \max\{F_{Y_1}(x), \ldots, F_{Y_a}(x)\}$. Stochastic set dominance is denoted in the following as $X \succeq_{SSD} S$. If the given condition is not fulfilled, this is denoted $X \not\succeq_{SSD} S$. Stochastic set dominance does not require but follows from first-order stochastic dominance, i.e., if $\exists Y \in S \mid X \succeq Y \implies X \succeq_{SSD} S$.*

Definition 10 (Independence). *Two random variables X and Y are independent if the pair of events $\{X = x\}$ and $\{Y = y\}$ are independent for all $x, y \in \mathbb{R}$. Formally,*

$$\mathbb{P}[X = x, Y = y] = \mathbb{P}[X = x]\mathbb{P}[Y = y], \quad \forall\, x, y \in \mathbb{R}.$$

Definition 11 (Convolution or sum of random variables). *If X and Y are independent random variables on $(\Omega, \mathcal{F}, \mathbb{P})$, then $Z = X + Y$ has probability density function, when X and Y are discrete random variables*

$$\mathbb{P}[Z = z] = \sum_{x=-\infty}^{\infty} f_X(x) f_Y(z - x), \quad \forall\, z \in \mathbb{Z}, \quad (1)$$

and for continuous random variables:

$$\mathbb{P}[Z = z] = \int_{-\infty}^{\infty} f_X(x) f_Y(z - x)\,\mathrm{d}x. \quad (2)$$

Lemma III.1 (Maximum between random variables). *If X and Y are independent random variables on $(\Omega, \mathcal{F}, \mathbb{P})$, then $Z = \max(X, Y)$ has cumulative probability function*

$$F_Z(z) = F_X(z) F_Y(z)$$

Proof. By definition of CDF, we have that:

$$\begin{aligned} F_Z(z) &= \mathbb{P}[\max(X, Y) \le z] = \mathbb{P}[X \le z \wedge Y \le z] \\ &= \mathbb{P}[X \le z]\mathbb{P}[Y \le z] \; X \text{ and } Y \text{ are independent} \\ &= F_X(z) F_Y(z). \end{aligned}$$

□

IV. Planning Methodology

This section gives a step-by-step description of the planning methodology. Sec. IV-A defines a feasible plan and identifies dependencies between robot tasks and human tasks. Sec. IV-B derives the stochastic duration of a plan or sub-sequence. Sec. IV-C introduces a B&B algorithm to search for candidate plans having the shortest durations in a stochastic sense, while Sec. IV-D describes risk-aware plan selection from this set. Sec. IV-E presents a pruning strategy based on stochastic dominance and proves this strategy is safe.

A. Plan feasibility and dependencies with human tasks

A plan is the set of all *robot* tasks in the RTSG model except those in non-selected alternative branches, ordered in a feasible sequence from the start node to the goal node. A feasible plan must fulfill the constraints imposed by the RTSG model (see Sec. III). For the model in Fig. 1, a feasible plan is exemplified by ($Start$, $FE2$, $DL1$, $RA2$, $MV1$, $PI1$, $FE1$, DL, $Goal$). It does not violate precedence or lock constraints. The completion of the plan depends on the human task $HA1$, which belongs to the selected alternative branch in the RTSG model starting with $DL1$. Due to precedence constraints, $HA1$ can not start, i.e., become *enabled*, until $DL1$ is completed. This makes $DL1$ the *enabling task* of $HA1$ in this plan. Due to the AND-JoinSync (&JS) node, $MV1$ is not considered completed until $HA1$ is completed. Therefore, depending on the outcome of task durations, the robot may need to wait for completion of $HA1$ after reaching $MV1$. This makes $MV1$ the *dependent task* of $HA1$. If a human task is enabled in a plan, there is always a dependent task, e.g., if the upper &JS node in Fig. 1 is replaced with an &J node, the goal node becomes the dependent task of $HA1$.

B. The duration of a plan

The duration of a plan is referred to as makespan. It depends on stochastic routing and task durations, $R_{\tau,\tau'}$, A_τ, B_h, of planned robot tasks and enabled human tasks. From these inputs, we derive the duration of a plan or sub-sequence as a random variable without restricting its probability distribution. To support this derivation, a few definitions are introduced: A plan is a sequence of robot tasks without element repetition, defined as $P_{0,n} = (\tau_0, \ldots, \tau_n) \subseteq T$ where τ_0 is the start state, τ_n the goal state and $\tau_1, \ldots, \tau_{n-1}$ are robot tasks. $P_{0,n}$ represents a feasible plan of the RTSG model, e.g., $\tau_0 = Start$, $\tau_1 = DL1$, $\tau_2 = RA2$, $\tau_3 = MV1$, etc. $P_{j,k}$ represents a sub-sequence from τ_j to τ_k, where the first task, τ_j, represents the start location and the following, $\tau_{j+1}, \ldots, \tau_k$, are robot tasks to be performed. For example, $P_{1,3}$ = (τ_1, τ_2, τ_3) starts at the location of τ_1 and thereafter performs τ_2 followed by τ_3. $\mathrm{HD}(P_{0,k}, \tau_i) \subseteq H$ represents the set of human tasks whose dependent task in $P_{0,k}$ is $\tau_i \in T$, $i \le k$. If a human task, $h \in H$, has an enabling task $\tau_p \in P_{0,k}, 0 \le p \le k$, then $\mathrm{EI}(P_{0,k}, h) = p$ represents the sequence index of the enabling task. Similarly, $\mathrm{DI}(P_{0,k}, h)$ represents the sequence index of the dependent task of h within $P_{0,k}$. The duration of a sub-sequence $P_{j,k}$ is indicated as $K_{j,k}$ and computed as:

$$K_{j,k} = \max(\mathcal{D}(j,k)) \quad (3)$$

where $\mathcal{D}(j,k)$ is a set of alternative durations between τ_j and τ_k and the $\max$ operation is defined in Lemma III.1. $\mathcal{D}(j,k)$ combines the robot's sequential routing and action durations with all possible combinations of waiting for human tasks. Recursion for $\mathcal{D}(j,k)$ is given by:

$$\begin{aligned} \mathcal{D}(j,k) =& (\mathcal{D}(j,k-1) + R_{\tau_{k-1,k}} + A_{\tau_k}) \cup \\ & \bigcup_{h \in \mathrm{HD}(P_{0,k}, \tau_k)} (\mathcal{D}(j, \mathrm{EI}(P_{0,k}, h)) + B_h) \end{aligned}$$

where $\mathcal{D}(j, v) = \emptyset$, $\forall v < j$. The base case $\mathcal{D}(j,j) = \{C^0\}$ represents a set with one random variable having a constant value of zero. The sum operator is defined in Def. 11. The sum of a set of durations $\mathcal{D}(j,k)$ and a duration X is element wise, i.e., $\mathcal{D}(j,k) + X = \{d + X : d \in \mathcal{D}(j,k)\}$. A human task may affect $K_{j,k}$ if the enabling and dependent tasks are included in the sub-sequence. Using Eq. 3 is not always the most efficient way to calculate the duration. For example, if a new task is appended to a sub-sequence with a previously known duration, a total re-computation is not always necessary. By exploiting the structure of a sequence's dependencies with human tasks, the duration can often be computed by adding the durations of consecutive sub-sequences, i.e., $K_{j,k} = K_{j,l} + K_{l,k}$, $j < l < k$. A sufficient condition for this sum rule is given in Eq. 4. The condition requires all human tasks enabled in $P_{j,l}$, either to have no dependent tasks in $P_{l+1,k}$, or to be completed latest before the start of τ_{l+1}:

$$\bigwedge_{h \in \mathrm{HE}(P_{j,l})} \Big(\mathrm{DT}(P_{l+1,k}, h) = \emptyset \ \vee p_0(K_{\mathrm{EI}(P_{0,l},h),l}) \geq p_{100}(B_h) \Big) \quad (4)$$

where $\mathrm{HE}(P_{j,k}) \subseteq H$ is the set of human tasks enabled in $P_{j,k}$. $\mathrm{DT}(P_{j,k}, h) \in T$ is the dependent task for $h \in H$ in $P_{j,k}$. $\mathrm{DT}(P_{j,k}, h) = \emptyset$, if the dependent task is elsewhere or h not is enabled. For later reference, $\mathrm{HD}(P_{j,k}) \subseteq H$ is the set of human tasks whose dependent tasks are in $P_{j,k}$.

C. Extended B&B algorithm

The goal of the planning algorithm is to identify the set of plans that reaches the goal state with a minimized makespan in a stochastic sense. Our algorithm extends a previously developed deterministic B&B algorithm [9] where a breadth-first forward search from the start state towards the goal state of an RTSG model is used to compute an optimized plan. Each search node represents a unique sub-sequence, $P_{0,i}$, with a duration, $K_{0,i}$, where i is the search depth. Children are identified by searching the RTSG graph for feasible selections of the next robot task. To limit the search tree growth, a pruning selection is made among two search nodes having the same depth, $P^A_{0,i}$ and $P^B_{0,i}$, if they are considered *equivalent*, i.e., they contain the same set of tasks, Eq. 5, and the last task is the same, Eq. 6. :

$$\{\tau : \tau \in P^A_{0,i}\} = \{\tau : \tau \in P^B_{0,i}\} \quad (5)$$

$$\tau^A_i = \tau^B_i \quad (6)$$

In essence, equivalent search nodes represent the same state, but reached with different sequences. Pruning should stop exploring the search node having the *longest duration*, i.e., the pruning selection criterion. However, this criterion is unclear when applied on random variables. A naive approach would be to replace *longest duration* with *longest k-th percentile of the duration*, e.g., $p_k(K_{0,i})$, where the risk-aware planner selects the targeted k. Unfortunately, our experiments confirm this criterion is unsafe and may stop exploring potentially better full-length plans for a given k. Instead, our pruning selection criterion is based on first-order stochastic dominance, proven safe in the targeted application types in Sec. IV-E. When comparing alternative equivalent sub-sequences during the tree exploration, one sub-sequence can *stochastically dominate* another (see Def. 8). This means it always has a higher chance of providing a longer duration, for any k, than the other and may therefore be pruned. The outcome of the tree exploration is a set of goal-reaching plans, E. From this set, a candidate set, $Q \subseteq E$, is identified, containing the plans that do not dominate the set of other plans:

$$Q = \{P \mid P \in E \wedge K \not\succeq_{SSD} \{K' \mid P' \in E \setminus P\}\} \quad (7)$$

where K and K' are the makespans of P and P', respectively. Using the proposed stochastic set dominance criterion ($\not\succeq_{SSD}$) in Eq. 7 instead of first-order stochastic dominance ($\not\succeq$) is more stringent, hence every plan in Q will have the lowest k-th percentile, p_k, of all plans in E for at least some k, which otherwise would not be guaranteed.

D. Risk aware plan selection

It is possible to specify a desired risk level $k \in [0, 100]$, where a lower value increases the risk. From this input, a candidate plan is automatically selected from Q, having the minimum makespan reachable with a probability of k, i.e., a plan with minimum p_k. Specifying a lower k reduces p_k and the probability of making it. Changing k may also lead to selecting a different candidate plan, which is optimal for the new k. To complement the risk level, a user decision may be supported by a visual comparison of PDF curves, indicating makespan variances of the candidate plans. $|Q| = 1$ implies there is a single plan having the highest chance of providing the shortest makespan for any risk level, i.e., a *risk independent optimal plan*

E. Safe pruning method

As previously mentioned, the B&B algorithm can prune nodes at the same search depth if considered equivalent. However, the equivalence conditions, Eqs. 5-6, are not sufficient when introducing human tasks, potentially running concurrently with the last robot task. If the search tree is further explored, children nodes will sometimes include a dependent task. For these nodes, the concurrent human task may affect the plan duration differently, depending on when the human task is enabled and when the dependent task occurs. This makes a pruning decision solely based on Eqs. 5-6 unsafe. To remedy this, one additional condition for equivalence, Eq. 8, is introduced. This condition is fulfilled if the influence of all enabled human tasks is fully accounted for in the search node durations, $K^A_{0,i}$ and $K^B_{0,i}$. If not, equivalence is still possible for some instances, starting with identical sub-sequences up to a point where enabling any human task also includes the corresponding dependent task in the remaining sequence. The condition above is expressed as:

$$\Bigg[\bigwedge_{h \in \mathrm{HE}(P^A_{0,i})} \Big(\mathrm{DT}(P^A_{0,i}, h) \neq \emptyset \ \vee p_0(K^A_{\mathrm{EI}(P^A_{0,i},h),i}) \geq p_{100}(B_h) \Big) \wedge \bigwedge_{h \in \mathrm{HE}(P^B_{0,i})} \Big(\mathrm{DT}(P^B_{0,i}, h) \neq \emptyset \ \vee p_0(K^B_{\mathrm{EI}(P^B_{0,i},h),i}) \geq p_{100}(B_h) \Big) \Bigg] \vee \Bigg[\mathrm{MI}(P^A_{0,i}) = \mathrm{MI}(P^B_{0,i}) \wedge \bigwedge_{j=0}^{\mathrm{MI}(P^A_{0,i})} \tau^A_j = \tau^B_j \Bigg] \quad (8)$$

$$\text{where } \mathrm{MI}(P_{0,i}) = \min\{l \mid \mathrm{HE}(P_{l+1,i}) = \mathrm{HD}(P_{l+1,i})\} \quad (9)$$

By definition, condition Eq. 8 is true with no human tasks.

Theorem IV.1. *Let $P^A_{0,i}$ and $P^B_{0,i}$ be equivalent sequences and $K^A_{0,i}$ dominate $K^B_{0,i}$ in terms of a first-order stochastic dominance (FSD), in short, $K^A_{0,i} \succeq K^B_{0,i}$. Then, the duration of any plan starting with sub-sequence $P^A_{0,i}$ dominates the duration of at least one plan starting with $P^B_{0,i}$.*

Proof. Assume $P^A_{0,n} = (\tau^A_0, \dots, \tau^A_i, \tau^A_{i+1}, \dots, \tau^A_n)$ is a feasible plan. We define $P^B_{0,n} = (\tau^B_0, \dots, \tau^B_i, \tau^A_{i+1}, \dots, \tau^A_n)$. Equivalence of $P^A_{0,i}$ and $P^B_{0,i}$ implies conditions Eqs. 5, 6 hold, i.e., the set of completed tasks and the last task τ_i are the same, representing the same planning state in the RTSG model. Therefore, by having an identical continuation as $P^A_{0,n}$ from τ_{i+1}, $P^B_{0,n}$ is also a feasible plan. The proof is segmented into two distinct cases based on fulfilling the sub-conditions specified in Eq. 8. In case the first sub-condition in Eq. 8 is true, it implies condition Eq. 4 is also true for $P^A_{0,i}$, $P^B_{0,i}$ and we can express the duration of corresponding plans as $K^A_{0,n} = K^A_{0,i} + K^A_{i,n}$ and $K^B_{0,n} = K^B_{0,i} + K^A_{i,n}$, respectively. Thereby, both $K^A_{0,n}$ and $K^B_{0,n}$ can be defined as a monotone, increasing and continuous function of $K_{0,i}$:

$$K_{0,n} = K_{0,i} + K^A_{i,n}$$

$K^A_{0,i} \succeq K^B_{0,i} \implies K^A_{0,n} \succeq K^B_{0,n}$ in accordance with the FSD-theorem in [20].

If the second sub-condition in Eq. 8 is true, it implies $P^B_{0,n} = (\tau^A_0, \dots, \tau^A_{\mathrm{MI}}, \tau^B_{\mathrm{MI}+1} \dots, \tau^B_i, \tau^A_{i+1}, \dots, \tau^A_n)$ where $\mathrm{MI} = \mathrm{MI}(P^A_{0,i}) = \mathrm{MI}(P^B_{0,i})$. By the definition of MI in Eq. 9, all human tasks enabled within $P_{\mathrm{MI},i}$ also have their dependent tasks within $P_{\mathrm{MI},i}$. Some human tasks may be enabled within $P^A_{0,\mathrm{MI}}$ and have their dependent tasks within $P^A_{i+1,n}$, thereby running concurrently with $P_{\mathrm{MI},i}$. All remaining human tasks are enabled and have their dependent tasks locally within $P^A_{0,\mathrm{MI}}$ or $P^A_{i,n}$. Considering this structure of human dependencies, the duration of plans A and B can be derived from Eq. 3 as:

$$K_{0,n} = \max\big((K^A_{0,\mathrm{MI}} + K_{\mathrm{MI},i} + K^A_{i,n}) \cup \bigcup_{\substack{h \in \mathrm{HE}(P^A_{0,\mathrm{MI}}) \cap \\ \mathrm{HD}(P^A_{i+1,n})}} (K^A_{0,\mathrm{EI}(P_{0,\mathrm{MI}},h)} + B_h + K^A_{\mathrm{DI}(P_{i+1,n},h),n})\big)$$

where $K_{\mathrm{MI},i}$ becomes included as a single summand in a single operand of the max operator. Thus, $K_{0,n}$ is a monotone (non-strictly) increasing and continuous function of $K_{\mathrm{MI},i}$. Additionally, the second sub-condition in Eq. 8 implies condition Eq. 4 so that:

$$K_{\mathrm{MI},i} = K_{0,i} - K^A_{0,\mathrm{MI}}$$

Thus, $K_{\mathrm{MI},i}$ is a monotone, increasing and continuous function of $K_{0,i}$. $K^A_{0,i} \succeq K^B_{0,i} \implies K^A_{\mathrm{MI},i} \succeq K^B_{\mathrm{MI},i} \implies K^A_{0,n} \succeq K^B_{0,n}$ in accordance with the FSD-theorem in [20]. □

The theorem suggests we can safely prune $P^A_{0,i}$, since the exploring of this search node will not result in a better plan than the best plans starting with $P^B_{0,i}$.

V. Evaluation

This section presents an experimental evaluation of the proposed planning approach, including a use case scenario for a planning problem, a deterministic benchmark approach, Monte Carlo simulations, followed by the evaluation results and their interpretation and a discussion of the outcomes.

Algorithm 1: Monte Carlo makespan computation

function ComputeMakespan($P_{0,n}$)
  $i \leftarrow 1$
  $D_0 \leftarrow 0$
  $B^{'}_h = B_h \quad \forall h \in \mathrm{HE}(P_{0,0})$
  while $i \leq n$ **do**
    $B = \max\{0, B^{'}_h \mid h \in \mathrm{HD}(P_{i,i})\}$
    $D_i = \max(D_{i-1} + R_{\tau_{i-1},\tau_i} + A_{\tau_i}, B)$
    $B^{'}_h = D_i + B_h \quad \forall h \in \mathrm{HE}(P_{i,i})$
    $i \leftarrow i + 1$
  return D_n

A. Use case scenario

A planning problem, use case A, is modeled with the RTSG model in Fig. 1, having two human assembly tasks at different stations, where the robot delivers parts and fetches completed assemblies. Alternatively, the robot can perform one or both assemblies at robot assembly stations. Additionally, there are two robot fetch tasks at different locations. Routing and task durations are modeled with uniform distributions. Modeling resolution is 0.1 s. Assuming humans are somewhat faster than robots, their expected duration is modeled to be slightly lower but with a higher variance. An extended version of the planning problem, use case B, has five additional tasks ($DL3$, $HA3$, $MV3$, $PI3$, $RA3$) by inserting one extra assembly branch between the AND-Fork and the AND-Join node pairs in the RTSG model.

B. Deterministic benchmark approach

The rationale of the benchmark approach is to provide a deterministic version of the probabilistic approach, highlighting differences in the outcome if using *non-stochastic* values, identical with the expected values of the stochastic approach, to model routing and task durations. The deterministic approach uses the same B&B algorithm but searches for a *single* plan with minimized makespan. Sequence durations are calculated in the same way as detailed in section IV-B, but using *non-stochastic* '+' and $\max$ operators. For pruning, it uses the same extended criteria for equivalence as detailed in Eqs. 5, 6 and 8, but with *longest duration* instead of *stochastic dominance* as pruning selection criterion. Since stochastic dominance does not always occur, the deterministic approach can generally prune more search nodes.

C. Monte Carlo simulations

Monte Carlo simulations are used to verify the correctness of makespans distributions computed by the incremental search tree exploration using Eq. 3 combined with the sum rule Eq. 4. For a single observation of a given plan, task and routing durations are generated randomly from their modeled distributions. The observed makespan is computed using the efficient Alg. 1, which is applicable for a full-length plan. By generating thousands of makespan observations, a probability distribution is derived by counting the number of observations that fall within different intervals. The interval length is 0.1 s and $5 \cdot 10^5$ simulations are run for each plan.

D. Evaluation results

At the top of Fig. 2, CDFs for the candidate plans of the stochastic approach are exemplified for use case A. The vertical axis indicates the k value of a percentile while the horizontal axis indicates the percentile, i.e., the maximum makespan for the best k share of outcomes. A risk-tolerant planner might prefer the red plan, which is the best plan with a 0-30% chance of making the corresponding percentile or better. On the other hand, the risk-averse planner might go for the black plan, which is the best plan with an 85-100% chance of making the corresponding percentile or better. The green plan is best with a 30-70% chance, while the blue is best between 70-85%. The second graph indicates corresponding PDFs, and their expected makespans as vertical lines, giving an intuition on possible variations. The third graph indicates the makespan generated by the deterministic approach as a solid vertical line. For comparison, the dotted lines show the PDF and the expected makespan (see Def. 2) of this plan for the stochastic approach. In this example, the deterministic plan is also among the stochastic candidate plans. The bottom graph provides Monte Carlo distributions of the plans, normalized by dividing interval counts with the total number of observations. Use case B is exemplified in the same way in Fig. 3. Here, the plan computed by the deterministic approach is not found among the candidate plans of the stochastic approach. This is because the deterministic plan stochastically dominates the set of all other feasible plans according to Def. 9, thereby excluding it from the candidate set of Eq. 7. In the top graph, the CDF of the deterministic plan is included as a dotted line. For every k-value, the percentile of the deterministic plan is higher than the percentile of at least one of the candidate plans of the stochastic approach. However, the deterministic plan does not dominate any candidate plan in terms of first-order stochastic dominance. In a statistical comparison of the planning approaches, 100 plans were computed for each use case with randomized task locations and intervals of

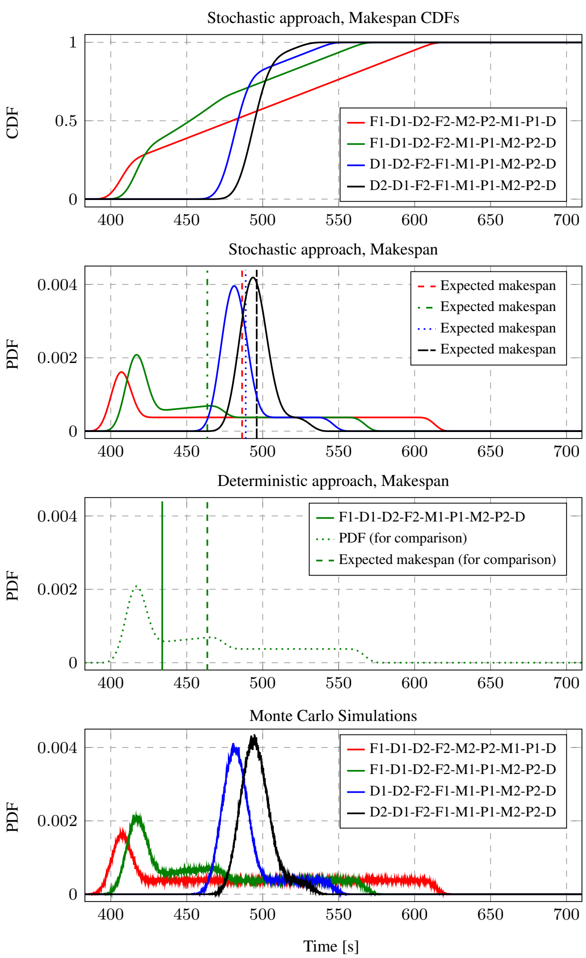

Fig. 2: Makespans for use case (A) with two assembly tasks.

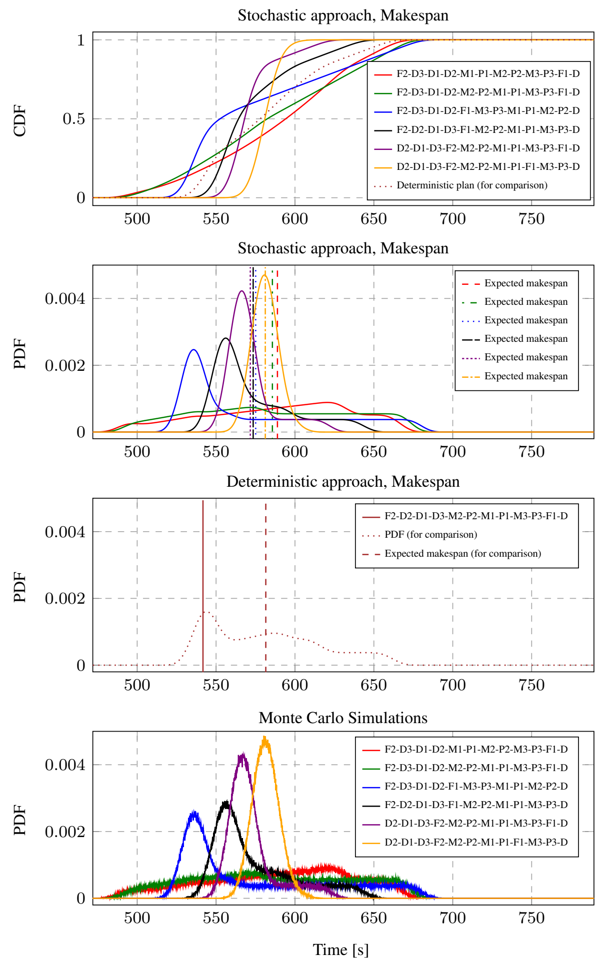

Fig. 3: Makespans for the extended use case (B).

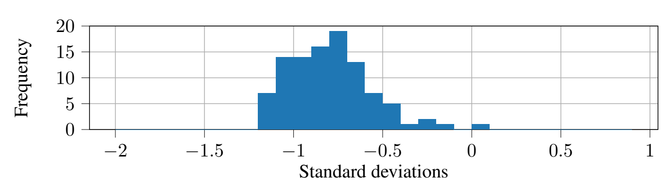

Fig. 4: Frequencies of deterministic makespan subtracted with expected stochastic makespan, for use case A.

Planning approach	Number of explored nodes		Number of pruned nodes		Among the best stochastic plans	
	A	B	A	B	A	B
Stochastic B&B	773	16995	251	6262	(100%)	(100%)
Deterministic B&B	594	12116	259	5104	65%	47%

TABLE I: Statistic comparison of the planning approaches.

input distributions. Table I presents the average number of explored and pruned nodes and indicates how many of the deterministic plans were found among the candidate plans of the stochastic approach. In Fig. 4, a histogram presents frequencies of makespans of the deterministic approach subtracted by expected makespans of the stochastic approach, expressed as several standard deviations, for use case A. A negative value means the deterministic makespan is shorter than the expected stochastic makespan.

E. Evaluation discussion

The Monte Carlo simulations in the bottom graphs of Figs. 2 and 3 are, if the noise is omitted, very similar to the PDFs in the corresponding figures, thereby supporting the correctness of the plan durations computed by the proposed planning approach. A large share of the PDFs are quite asymmetric, highlighting the advantage of not limiting the type of probabilistic distribution that can be represented. In this study, the deterministic approach tends to underestimate the makespan, as indicated in Fig. 4 and exemplified in the 3rd graph of Figs. 2, 3. This tendency magnifies the inherent problem of the deterministic approach, where a computed plan is associated with a more or less unknown risk. The stochastic approach considers this risk while searching for the best plans, providing information on how much a makespan can vary and suggesting the best plan with respect to the planner's willingness to risk. The deterministic plan is not always among the best plans of the stochastic approach, here in 65% and 47% of the runs (Table I). This highlights the risk of computing an inferior plan by not considering uncertainties. The stochastic approach needs to explore 30% and 40% more nodes due to the safe but more restrictive pruning selection criterion. Still, the potential for pruning in the proposed approach is significant.

VI. Conclusion

This paper proposes a novel methodology to compute an optimized collaborative plan while considering uncertain task durations and the risk willingness of a human planner. Relevant planning problems modeled with a Robot Task Scheduling Graph (RTSG) accommodate uncertainties by representing them as random variables. These are effectively tackled using a Branch-and-Bound (B&B) algorithm, incorporating a safe pruning strategy grounded in first-order stochastic dominance. The result is a set of the best plans for all risk levels, with makespans represented as probability distributions, empowering planners to make informed decisions based on their situational risk tolerance. Future research includes the exploration of techniques for learning input distributions dynamically. Additionally, we foresee extending our methodology to consider other types of risks and handle more complex scenarios, such as centralized or decentralized multi-agent task allocation, agent load restrictions and intricate dependencies between tasks.

References

[1] A. Ameri E., B. Miloradovic, B. Çürüklü, A. V. Papadopoulos, M. Ekström, and J. Dréo. Interplay of Human and AI Solvers on a Planning Problem. In *IEEE Int. Conf. Sys., Man, & Cyb. (SMC)*, 2023.

[2] A. Casalino and A. Geraci. Allowing a real collaboration between humans and robots. *Special Topics in Information Technology*, 2021.

[3] A. Casalino, E. Mazzocca, M. G. Di Giorgio, A. M. Zanchettin, and P. Rocco. Task scheduling for human-robot collaboration with uncertain duration of tasks: a fuzzy approach. In *Int. Conf. Control, Mechatronics and Automation (ICCMA)*, pages 90–97, 2019.

[4] K. Darvish, B. Bruno, E. Simetti, F. Mastrogiovanni, and G. Casalino. Interleaved online task planning, simulation, task allocation and motion control for flexible human-robot cooperation. In *IEEE Int. Symp. Robot and Human Interactive Comm. (RO-MAN)*, pages 58–65, 2018.

[5] G. Ferguson, J. F. Allen, B. W. Miller, et al. TRAINS-95: Towards a mixed-initiative planning assistant. In *Conf. Artificial Intelligence Planning Systems (AIPS)*, pages 70–77, 1996.

[6] L. Johannsmeier and S. Haddadin. A hierarchical human-robot interaction-planning framework for task allocation in collaborative industrial assembly processes. *IEEE RA-L*, 2(1):41–48, 2016.

[7] J. Kinugawa, A. Kanazawa, S. Arai, and K. Kosuge. Adaptive task scheduling for an assembly task coworker robot based on incremental learning of human's motion patterns. *IEEE RA-L*, 2(2):856–863, 2017.

[8] A. Lager, A. V. Papadopoulos, G. Spampinato, and T. Nolte. A task modelling formalism for industrial mobile robot applications. In *Int. Conf. Advanced Robotics (ICAR)*, pages 296–303, 2021.

[9] A. Lager, G. Spampinato, A. V. Papadopoulos, and T. Nolte. Task roadmaps: speeding up task replanning. *Frontiers in Robotics & AI*, 9:816355, 2022.

[10] M. Lippi and A. Marino. A mixed-integer linear programming formulation for human multi-robot task allocation. In *IEEE Int. Symp. RO-MAN*, pages 1017–1023, 2021.

[11] Z. Liu, Q. Liu, L. Wang, W. Xu, and Z. Zhou. Task-level decision-making for dynamic and stochastic human-robot collaboration based on dual agents deep reinforcement learning. *Int. Journal of Advanced Manufacturing Tech.*, 115(11-12):3533–3552, 2021.

[12] P. Lou, Q. Liu, Z. Zhou, H. Wang, and S. X. Sun. Multi-agent-based proactive–reactive scheduling for a job shop. *Int. Journal of Advanced Manufacturing Tech.*, 59:311–324, 2012.

[13] E. Matheson, R. Minto, E. G. Zampieri, M. Faccio, and G. Rosati. Human–robot collaboration in manufacturing applications: A review. *Robotics*, 8(4):100, 2019.

[14] R. Müller, M. Vette, and O. Mailahn. Process-oriented task assignment for assembly processes with human-robot interaction. *Procedia CIRP*, 44:210–215, 2016.

[15] A. W. Palmer, A. J. Hill, and S. J. Scheding. Methods for stochastic collection and replenishment (scar) optimisation for persistent autonomy. *Robotics and Autonomous Systems*, 87:51–65, 2017.

[16] A. W. Palmer, A. J. Hill, and S. J. Scheding. Modelling resource contention in multi-robot task allocation problems with uncertain timing. In *IEEE Int. Conf. ICRA*, pages 3693–3700, 2018.

[17] R. Y. Rubinstein and D. P. Kroese. *Simulation and the Monte Carlo method*. John Wiley & Sons, 2016.

[18] M. Saint-Guillain, T. Vaquero, S. Chien, J. Agrawal, and J. Abrahams. Probabilistic temporal networks with ordinary distributions: Theory, robustness and expected utility. *Journal of Artificial Intelligence Research*, 71:1091–1136, 2021.

[19] M. Shaked and J. G. Shanthikumar, editors. *Univariate Stochastic Orders*, pages 3–79. Springer New York, New York, NY, 2007.

[20] E. Wolfstetter, U. Dulleck, R. Inderst, P. Kuhbier, and M. Lands-Berger. *Stochastic dominance: theory and applications*. Humboldt-Univ., Wirtschaftswiss. Fak., 1993.

[21] M. Yang, E. Yang, R. C. Zante, M. Post, and X. Liu. Collaborative mobile industrial manipulator: a review of system architecture and applications. In *Int. Conf. Autom. & Comp. (ICAC)*, pages 1–6, 2019.

[22] A. M. Zanchettin, A. Casalino, L. Piroddi, and P. Rocco. Prediction of human activity patterns for human–robot collaborative assembly tasks. *IEEE Trans. Industrial Informatics*, 15(7):3934–3942, 2018.