Masters Thesis

A Formal Approach to Human Robot Collaborative Assembly Planning under Uncertainty

by

Momina Rizwan

Submitted to the Graduate School of Engineering and Natural Sciences in partial fulfilment of the requirements for the degree of Master of Science

Sabancı University

Faculty of Engineering and Natural Sciences

Mechatronics Engineering

January 2019

A Formal Approach to
Human Robot Collaborative Assembly Planning
under Uncertainty

APPROVED BY

Assoc. Prof. Dr. Volkan PATOĞLU ....
(Thesis Supervisor)

Assoc. Prof. Dr. Esra ERDEM PATOĞLU ....
(Thesis Co-Supervisor)

Asst. Prof. Dr. Reyyan YENİTERZİ ....

Asst. Prof. Dr. Öznur TAŞTAN OLCAN ....

Asst. Prof. Dr. Bekir BEDİZ ....

DATE OF APPROVAL: 07/01/2019

Acknowledgements

I would like to express my sincere gratitude to my thesis advisors Assoc. Prof. Dr. Volkan Patoğlu and Assoc. Prof. Dr. Esra Erdem Patoğlu for their support, and guidance. Without their continuous assistance and encouragement, this work have not been possible. I am greatly thankful for all the opportunities they have provided me. I learned the process and principles of doing research. It was a great opportunity for me to work under such great professors.
Furthermore, I would like to extend my thanks to the lab members of Cognitive Robotics Lab for their precious feedback. I would specially like to thank Ibrahim Faruk Yalciner for his help in understanding HCP-ASP which is the basis of my thesis.I would also like to express my regards to the jury members for their time and feedback.
Finally, I must express my very profound gratitude to God Almighty, and my family for providing me with unfailing support and encouragement throughout my years of study. I would like to specially thank my husband Faseeh Ahmad for his co-operation and support. This accomplishment would not have been possible without them. Thank you.

ABSTRACT

A FORMAL APPROACH TO HUMAN-ROBOT COLLABORATIVE ASSEMBLY PLANNING UNDER UNCERTAINTY

Momina Rizwan

Mechatronics Engineering, Master of Science, 2019

Thesis Supervisor: Assoc. Prof. Dr. Esra Erdem Patoğlu
Thesis Supervisor: Assoc. Prof. Dr. Volkan Patoğlu

Keywords: Assembly Planning, Hybrid Conditional Planning, Human Robot-Interaction, Collaborative Assembly Planning.

For assembly planning, robots necessitate certain cognitive skills: high-level planning of actuation actions is needed to decide for their order, while geometric reasoning is needed to check their feasibility. For collaborative assembly tasks with humans, robots require further cognitive capabilities, such as commonsense reasoning, sensing, and communication skills, not only to cope with the uncertainty caused by incomplete knowledge about the humans behaviors, but also to ensure safer collaborations.

We introduce a novel formal framework for collaborative assembly planning under uncertainty that utilizes hybrid conditional planning extended with commonsense reasoning and a rich set of communication actions for collaborative tasks. We show the applicability of our approach over a furniture assembly domain, where a bi-manual Baxter robot collaborates with a human to assemble a table, with dynamic simulations and physical implementations. We also evaluate our approach experimentally in this domain with respect to quantitative and qualitative performance measures.

Contents

List of Figures

List of Tables

Chapter 1

1 Introduction

While high scale industries are moving towards customized products, robotic assembly tasks are becoming not only physically challenging, but also mentally challenging. For this reason, past few years have brought drastic changes to industrial robotics. Previously, working areas of humans and robots were strictly separated, but now there is need for the robots to collaborate with humans, as flexible assembly systems require both the precision of robots and the dexterity of humans. Furthermore an effective and socially appropriate human-robot interaction may lead to better work performance and team satisfaction while ensuring safety. The involvement of human in the robot workplace, however, poses further challenges due to uncertainty about the actions, behaviors and intentions of human.

1.1 Challenges

Collaborative assembly planning to produce a customized product necessitates robots to possess certain cognitive capabilities.

For instance, for assembly planning, high-level task planning is required to decide the order of actuation actions (e.g., picking, holding, joining, placing), while sensing is required to resolve uncertainty due to incomplete knowledge about the world (e.g., to check which table legs have round end points and thus can be assembled to the table top). Meanwhile, geometric reasoning is required to check the feasibility of both actuation and sensing actions (e.g., checking whether the robot can reach the table leg without any collisions to ensure feasibility of pick action). For collaborations with humans, robots need to be furnished with further cognitive capabilities, including commonsense reasoning (e.g., knowing that humans cannot carry heavy parts), sensing to resolve uncertainty about human actions (e.g., checking whether the human is holding the table leg to be assembled), and communication skills to resolve uncertainty about human intentions and to ensure safe and socially acceptable interactions. These communication skills involve greetings, asking/offering help, confirming intentions, requesting actions, warnings, and providing explanations. Endowing robots with such a variety of cognitive capabilities make collaborative assembly planning even more challenging.

1.2 Contributions

We propose to address these challenges of collaborative assembly planning by a novel formal framework based on hybrid conditional planning (HCP) and Answer Set Programming (ASP) [55].

1. Hybrid conditional planning (HCP) allows planning of sensing actions in addition to actuation actions, based on their formal logical descriptions. It also embeds continuous geometric feasibility checks directly into logical action descriptions. In this thesis, we propose to solve assembly planning using HCP by modeling sensing actions to resolve the uncertainty caused by the incomplete knowledge about the world state. For that, we introduce a method to represent actuation actions and sensing actions in the logic formalism of Answer Set Programming.

2. We extend our approach to *collaborative* assembly planning where human is involved, by modeling communication actions to resolve the uncertainty caused by the incomplete knowledge about the human physical and mental states. Embedding communication in planning is advantageous, not only for providing evidence-based explanations to humans, but also for safer collaborations. To model communication actions in the formal language of ASP,

we introduce a representation methodology depending on the type of communication action, the level of safety and the level of verbosity. According to this methodology, communication is initiated when needed, avoiding e.g., unnecessary calls for human help. Depending on the verbosity level, the robot can provide explanations. For instance, if the verbosity level is high the robot can tell that it cannot reach the assembly part as an explanation for its request for help. Depending on the safety level, the robot can offer help. For instance, if the safety level is high, the robot can offer help when the human attempts to pick a sharp object.

3. We further extend our approach to utilize a variety of commonsense knowledge by modeling it in the formalism of ASP and embedding it into descriptions of actuation, sensing and communication actions (e.g., greetings/acknowledgements before/after tasks, not getting too close to humans for safety purposes), as well as state constraints (e.g., a stable table has legs of the same size).

4. We have performed extensive experimental evaluations of the proposed approach and tested the practicality of our framework using dynamic simulations and human subject experiments.

In these studies, we have used the parallel hybrid conditional planner HCP-ASP [55] to compute the plans executable by the robot.

1.3 Outline

The outline of the remaining chapters, and a brief summary of each chapter are provided below:

- Chapter 2 discusses the literature review related to hybrid assembly planning and human-robot collaboration for assembly tasks. We also compare our approach with the most related works in the literature.

- Chapter 3 reviews the relevant preliminaries about Answer Set Programming, and hybrid conditional planning using HCP-ASP.

- Chapter 4 discusses our proposal to solve hybrid assembly planning using HCP by modeling of actuation, and sensing actions in the formalism of ASP. For feasibility checks, we use the state of the art RRT* motion planner from the OMPL library.

- Chapter 5 illustrates the applicability of our approach over the assembly of a table by a bi-manual Baxter robot. We also present some experimental results to discuss its scalability.

- In Chapter 6, we extend our hybrid assembly planning approach to include collaborations with humans, by formally modeling communication actions and by embedding relevant commonsense knowledge and feasibility checks into their descriptions.

- In Chapter 7, we illustrate a practical application of our approach over the assembly a table by a bi-manual Baxter robot collaboratively with a human team-mate. We test the case study in a dynamically simulated environment and present empirical results.

- Chapter 8 discusses the results of some physical experiments with human subject to show that collaborations are efficient, safe and natural.

- Chapter 9 conclude with a summary of the contributions and potential directions for future research.

Chapter 2

2 Literature Review

This work focuses on the human-robot collaborative assembly planning in the presence of uncertainty due to incomplete information about the world states, and human belief states. In Section 2.1, we discuss related works that study the challenges in hybrid task and motion planning. Section 2.3, we review the approaches employed to solve the TAMP problems for assembly planning. Section 2.4 describes the recent approaches proposed to deal with different challenges related to collaborative assembly planning.

2.1 Hybrid Task and Motion Planning (TAMP)

Combining task planning and motion planning (TAMP) for manipulation planning has been studied using different methods, e.g., with search-based approaches (based on systematic search over hybrid states) [4, 22, 25] and logic-based approaches (based on formal representations of hybrid actions) [6, 10, 21]. Some studies on TAMP in service robotics have considered uncertainty due to incomplete knowledge, e.g., by belief-state planning including sensing actions [25], while others have utilized commonsense knowledge, e.g., by logic-based knowledge representation methods [11].

2.2 Planning Under Uncertainty

Artificial intelligence literature offers several approaches to deal with uncertainty. The available approaches used to deal uncertainty due to incomplete knowledge about the dynamic environment are as follows:

- In policy generation, uncertainty is encoded as probability distributions. In this approach partial observability is considered and state-action pairs to maximize rewards are generated. In this approach, state-action pairs may not reach a goal.

- In conformant planning, uncertainty is encoded as sets of states. No observability, i.e. no sensing, is considered and a sequence of actions to reach a goal is generated. In this conservative approach, there exists no guarantee to reach the goal.

- In conditional planning uncertainty is encoded as sets of states. Partial observability is considered and a tree of action sequences to reach a goal under all possible contingencies is generated.

In conditional planning [40, 43, 54], actuation actions are modeled as deterministic actions and sensing actions as nondeterministic actions, and a tree consisting of sequences of these two types of actions is generated at the output. To make this idea more applicable to robotic domains, hybrid conditional planning (HCP) [55] further embeds geometric reasoning into descriptions of these actions.

2.3 Assembly Planning Problem

Research has been performed in many applications of assembly planning such as automobile [56] [23] and aircraft manufacturing industries [5], in furniture manufacturing industry [31], in the construction [35], and in nano-manufacturing [34].

Assembly planning problem has many levels: assembly sequence planning only deals with the geometric constraints of freely moving objects; assembly manipulation planning problem requires analysis of constraints arising from the task and object geometry as well as constraints arising from the specific configuration of the robot system. In this work, we focus on the hybrid assembly manipulation planning which aims to find an optimal sequence of assembly actions that are feasible to be executed by robots for motion planning. The input to the planner includes a set of assembly parts P, their initial configuration and the required goal conditions.

During the early work on assembly sequence planning [1], the precedence constraints are implicitly expressed as geometric relationships, which in the absence of collision-free trajectories allows two sub-assemblies to contact. Research has also been conducted on the automatic analysis of directional and non-directional-blocking graphs [28], or on geometrical information obtained with the analysis of the motion space [20]. Later, Liu [33] developed a task grammar that takes into account the fundamental principle on how the sequence of robot actions should be ordered and how a high-level task can be effectively decomposed into low-level operations by qualitative heuristics that guide through the geometric constraints of manipulation. Thomas and Torras [50] used spatial constraints to infer feasible assemblies. They proposed to search over possible configurations of parts that are consistent with feature set mappings and evaluate of the kinematic consistency of an assembly.

More recent works provide whole automated system to generate assembly plans given the goal state of the parts to be assembled. For instance, Thomas and Wahl [51] propose an approach that uses CAD-models, symbolic spatial relations and a robot work cell description as input. The goal state of the parts to be assembled is defined by the user interface. The generated assembly plans can be executed by robots by means of a set of predefined skill primitives. Another fully automated system for automatic assembly of aluminum profile constructions has been implemented in [38] which includes an assembly sequence planner integrated with a grasp planning tool, a knowledge-based reasoning method, a skill-based code generation, and an error tolerant execution engine.

In automated manufacturing, assembly plans aim to determine the proper order of assembly operations to build a coherent object. In the above mentioned approaches to assembly planning, the goal configuration is well-defined whereas our approach tends to search for such a goal state that satisfies some goal conditions and plan for actions to reach such a goal configuration.

There are some assembly planning approaches which use only motion planning rather than task and motion planning, for instance Kim et al. [29] propose a manipulation planning algorithm by implementing a Rapidly-exploring Random Trees (RRT) to generate the assembly path and the re-grasping path in different ways to obtain the manipulation path of the dual-arm robot for assembly task.

The notion of "assembly-by-disassembly principle", proposes that precedence is equivalent to blocking relationships between parts. In this approach, the problem of generating the assembly sequences is transformed into the problem of generating disassembly sequences in which the disassembly tasks are the inverse of feasible assembly tasks [7].

The most related work is conducted by Knepper et al. [31]. They used A Better Planning Language (ABPL) for representation of assembly planning problem and the PDDL planner. This approach does not support external program calls to incorporate non-symbolic feasibility checks and integration of common sense knowledge into the planning process. For example, calling a motion planner to determine if it is feasible to attach two sub-assemblies.

While all of the above studies have provided important contributions to the field of assembly planning, all of them considered the assembly planning problem in a tightly controlled and highly structured industrial environment. None of these works consider uncertainty in the world state while planning for an assembly task. Although these approaches use either a task planner or motion planner to solve assembly problem, they do not integrate those two approaches to perform hybrid planning. Unlike these related works, our framework provides a tight integration between high-level task planning and low-level feasibility checks to plan for feasible sequence of actuation and sensing actions. Furthermore, HCP considers uncertainties in the domain and plans for all possible contingencies.

2.4 Collaborative Assembly Planning

In typical assembly planning, no human-robot interaction is considered and uncertainties may exist only due to the incomplete knowledge of the world. However, human-robot collaboration is concerned with the uncertainty not only due to the incomplete knowledge about the state of the world but also due to the incomplete information about humans' actions, behavior, intentions, belief and desires.

To reveal knowledge about the humans' mental state, communication is necessary. Human-robot communications have been used to guide collaborative planning, before the planning takes place, or after planning,, that is, during the execution of the plan. For instance, in [30], communication between human and robot takes place before planning at a strategic level. While planning, they consider user's preferences to guide the planner. Experiments have been conducted in [52] where human-robot communication takes place during the execution of fetch and deliver tasks. This study compares the performance of human while robot assistants help the worker, who is assembling a part, by fetching and delivering components. The work in [32] focuses on the motion level robot adaptation for safe close proximity human-robot collaborative assembly tasks.

Our approach is different from the above mentioned approaches, as we consider communication actions while planning for collaborative task. It is desirable to ensure task fluency, as we do not need to re-plan according to human behaviors and intentions since we plan for each possible communication contingency beforehand. It is also preferable because for each planned communication, we can provide evidence based explanations.

2.4.1 Scheduling for Human-Robot Collaboration

Studies [17, 18] focus on scheduling tasks for human-robot teams rather than planning. They discuss the role of incorporating human preferences while scheduling a team task. Human-subject experiments have been conducted to understand how to best incorporate the human teammates' preferences in the team's schedule for safe and efficient human-robot coordination in time and space. Human subjects communicate their preferences before scheduling, the robot then schedules the team tasks taking these preferences into account. The problem of co-optimizing agent placement with task assignment and scheduling for large-scale multi-agent coordination under temporal and spatial constraints is studied in [57]. The problem is formulated as a multi-level optimization problem and solved with a multi-abstraction search approach. Study [41] focuses on real-time target prediction of human reaching motion and presents an algorithm based on time series classification. In their following work [53], a human-aware robotic system is presented that incorporates both predictions of human motion and planning in time to execute efficient and safe motions during automotive assembly tasks.

The research studies discussed above consider scheduling in time and space, while the focus of this work is planning under uncertainty for collaborative tasks. However, similar to our work, these studies make use of human preferences for safe and effective collaboration. In our proposed method, we can also embed human preferences about the verbosity and safety levels into our planning framework.

2.4.2 Policy Generation

Recall that policy generation [26] provides an alternative solution for planning under uncertainty as discussed in Section 2.2. In [19], communication is considered to resolve uncertainty while learning rewards for collaborative tasks. The actions are learned and policies are generated instead of conditional plans. A formal mathematical model of adaptation during human-robot collaboration is presented in [36], which discusses different ways that probabilistic planning and game-theoretic algorithms can enable reasoning over the uncertainty in robotic systems that collaborate with people. Later in [37], a formalism is proposed for combining verbal communication with actions towards task completion, in order to enable a human team-mate to adapt to its robot counterpart in a collaborative task. The formalism models the human adaptability as a latent variable in a mixed-observability Markov decision process. This work identifies two types of communication: verbal commands and state-conveying actions.

In comparison to these works, we provide a framework using HCP to model a richer set of verbal communication as part of the planning process. Moreover, our approach plans for hybrid actuation, hybrid sensing and hybrid communication actions which ensure the plan to be feasible during execution.

2.4.3 Metrics to Analyze Human-Robot Collaboration

Common metrics to guide the design and to evaluate the performance of human-robot systems have been proposed in [46]. This study discusses parameters such as reliability, efficiency, and risk to humans for human-robot systems operating in a hostile environment. It is discussed that in the context of human-robot systems, an intervention is not only driven by component failures, but includes many other factors that can make a robotic agent to request or a human agent to provide intervention. In [44] the effect of the nature of tasks, e.g. mental or physical challenge level of a task on the preference of participants for different interaction styles is studied. The goal is to determine the specific situations in which different interaction styles are most preferred.

These studies are in connection with our human-subject experiment, since we also utilize quantitative and qualitative measures in the spirit of [30] to evaluate the efficiency of our framework, by means of surveys applied to a diverse group of volunteers.

2.4.4 Dialog Planning

Human-robot interactions in natural language have been investigated by dialog-based approaches [16, 42, 49]. Some of these approaches use conditional planning [42], some use branching plans [45], and some use policy generation [19] to incorporate communication actions in plans to obtain further knowledge. For instance, Petrick and Foster [42] and Giuliani et al. [16] consider queries to learn what type of drink the human wants so that the robot prepares the customer's order accordingly. In their approach, human does not perform any actions that can change the world state. Sebastiani et al. [45] consider queries to negotiate which tasks will be performed by the robot or the human. In this work, negotiation actions are not formalized as nondeterministic actions as part of the domain description, and thus the contingencies in communications are generated by an algorithm as execution variables. In [19], authors consider queries to reduce state estimation uncertainty in policy generation. Their goal is to assist the human rather than to plan for completion of a task collaboratively. Different from these related work, our goal is to plan for collaborative actions, and we consider a richer set of communication tasks. We formalize all the communication actions as part of the domain description, and utilize them as part of conditional planning.

Studies [16, 42] are most related to our work, because communication actions are modeled formally as sensing actions and utilized while planning, for the purpose of constructing a dialogue: the robot communicates with human and serves them the requested drink. Our proposed approach utilizes communication for collaborative hybrid planning where human and robot perform actuation actions to reach a common goal and are aware of each other's intentions through observation and verbal communication. Collaborative tasks require richer communication actions, as observed above. Also, the representation language we use allows us to formalize commonsense knowledge.

The research work on Heirarchical Agent-based Task Planner (HATP) extended in [45] to generate conditional plans for human-robot collaborations by adding on-line negotiations is also closely related to our approach. In this work, they generate shared plans including sensing actions for human-robot interactions and collaborative actions. Our method does not negotiate on-line at every step of the task by asking who is going to perform which task but computes an off-line hybrid conditional plan before execution.

In particular, we compute a hybrid conditional plan for actuation, sensing , and communication actions and perform those actions only when needed. For instance, while executing a task, if the robot senses that human pro-actively takes an initiative for a task, it confirms human intention, otherwise it continues performing its own task. If the robot is unable to perform a task (verified via a feasibility check), it can ask help from the human team-mate. Human preferences may change from person to person, hence, due to this, we allow for specifying safety and verbosity level of plans to be generated.

Chapter 3

3 Preliminaries

Conditional planning, also known as contingent planning, enables planning from an initial state to a goal state in the presence of incomplete knowledge and partial observability [40, 43, 54] by considering all possible contingencies. Thus the plans (called conditional plans) are trees of actuation actions, whose effects are deterministic, and sensing actions, whose effects are non-deterministic, where each branch of the tree from the root to a leaf represents a possible execution of actuation and sensing actions to reach a goal state from the given initial state.

The existence of a conditional plan is an intractable problem: for polynomially bounded plans with partial observability, it is PSPACE-complete [2]. Despite this fact, there are various conditional planners. However, only few of them allows hybrid planning. In our studies, we use the hybrid conditional planner HCP-ASP [55].

The planner HCP-ASP is a compilation-based conditional planner: it transforms hybrid conditional planning into answer set computation. Therefore, the initial state, goal conditions and the action descriptions presented to HCP-ASP are in the formalism of Answer Set Programming (ASP) [55]. The hybrid conditional plans are computed using the ASP solver Clingo [13].

3.1 Answer Set Programming (ASP)

Answer Set Programming (ASP) [3] is a form of declarative programming paradigm oriented towards solving combinatorial search problems, such as planning. ASP is based on the stable model semantics of logic programming. The idea of ASP is to represent knowledge (e.g., actions of robots) as a program and to reason about the knowledge (e.g., find a plan of robots actions) by computing models, called answer sets [15], of the program using special implemented systems, called ASP solvers such as **iclingo** [14], dlvhex [9].

3.1.1 Programs

We consider ASP programs (i.e., nondisjunctive HEX programs [8]) that are sets of rules of the form

$$\mathit{Head} \leftarrow A_1, \ldots, A_m, \mathit{not}\ B_{m+1}, \ldots, \mathit{not}\ B_n$$

where $n \geq m \geq 0$, *Head* is a literal (a propositional atom p or its negation $\neg p$) or $\perp$, and each A_i is an atom or an external atom. A rule is called a *fact* if $m = n = 0$, and a *constraint* if *Head* is $\perp$.

Note that there are two sorts of negation: classical negation $\neg$ as in classical logic, and default negation *not*. Intuitively, $\neg p$ means that "it is known that p is not true" whereas *notp* means that "it is not known that p is true". Default negation is useful in expressing default values of atoms.

An external atom is an expression of the form $\&g[y_1, \ldots, y_k](x_1, \ldots, x_l)$ where $y_1, \ldots, y_k$ and $x_1, \ldots, x_l$ are two lists of terms (called input and output lists, respectively), and $\&g$ is an external predicate name. Intuitively, an external atom provides a way for deciding the truth value of an output tuple depending on the extension of a set of input predicates. External atoms allow us to embed results of external computations into ASP programs. They are usually implemented in a programming language of the user's choice, like Python, Lua.

For instance, the following rule:

$$\begin{gathered}\perp \leftarrow \mathit{place}(a, x_1, y_1, t), \mathit{holding}(a, o, t), \\ \mathit{not}\ \&\mathit{collision\_free}[a, x_1, y_1]()\end{gathered}$$

is used to express that, at any step t of the plan, a robot cannot place an object o at location (x_1, y_1) if there is no collision-free trajectory between them. The external atom $\&$*collision_free*$[a, x_1, y_1]()$ takes a, x_1, y_1 as inputs to the external computation (e.g., a Python program) that calls a motion planner (e.g. PRM, RRT, EST, SBL e.t.c.) to check the existence of a collision free trajectory for the arms a current co-ordinates to x_1 to y_1, and then returns the result of the computation as a precondition.

ASP provides special constructs to represent a variety of knowledge. For instance it is possible to express nondeterministic choice in ASP using “choice expressions with “cardinality constraints. Choice expressions help us to model occurrences and non-occurrences of actions. For instance, the following ASP rule

$$\{\mathit{sense}(at(o), t)\}$$

expresses that the action of sensing the location of an object can occur any time. Choice expressions with cardinality constraints help us to model nondeterministic effects of sensing actions. For instance, the following ASP rule

$$1\{\mathit{at}(o, l, t+1) : loc(l)\}1 \leftarrow \mathit{sense}(at(o), t)$$

describes that if sensing is applied to check the location of an object o (i.e., $\mathit{sense}(at(o), t)$), then we know that the object o is at one of the possible locations l; here, the location l is nondeterministically chosen by the ASP solver. Fourth, it is possible to express “unknowns using “cardinality expressions; e.g., the rule

$$\neg \mathit{at}(o, m, t) \leftarrow \{\mathit{at}(o, l, t) : loc(l)\}0$$

expresses that if objects location is not known (i.e., $\{\mathit{at}(o, l, t) : loc(l)\}0$) then it definitely can not be at a robots hand m. Fifth, we can express “weak constraints to minimize, e.g., the number of sensing actions:

$$\stackrel{\sim}{\leftarrow} \mathit{senseAct}(t)\ [2@2, t].$$

Finally, the incremental computation of an answer set by an ASP solver, like **iclingo** [14], allows for minimization of makespans (i.e., lengths) of plans.

3.2 Hybrid Conditional Planning

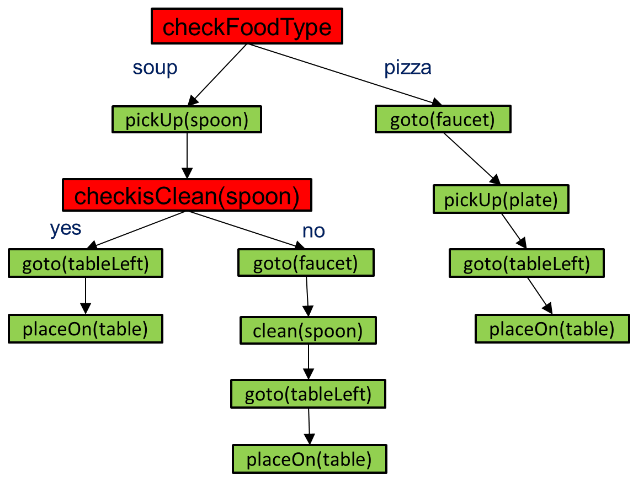


Figure 1: A sample hybrid conditional plan

A hybrid conditional plan can be identified as a labeled directed tree (V, E) as in Figure 1 where every branch represents a possible executable plan. The set $V = V_a \cup V_s$ of vertices denote actions in the conditional plan consisting of two types of vertices. The vertices in V_a represent hybrid actuation actions (e.g., the robot's navigation and manipulation actions integrated with feasibility checks) are highlighted as green in Figure 1. Whereas the vertices in V_s represent sensing actions or information gathering actions in general (e.g., sensing or checking out the location of an object) highlighted as red in Figure 1. The branching occurs when there is a sensing action with non-deterministic outcome, so every vertex in V_s has at least two outgoing edges while every vertex in V_a has a single outgoing edge while, each vertex in V_a has at most one outgoing edge based on the assumption that the actuation actions are deterministic. Each sensing action may lead to different outcomes/observations.

The set of edges E represents the order of actions in the directed graph. Let us denote by E_s the set of outgoing edges from vertices in V_s. Then a labeling function maps every edge (x, y) in E_s by a possible outcome of the sensing action characterized by x.

Given an initial state, goal conditions, and descriptions of actuation and sensing actions, hybrid conditional planning asks for a hybrid conditional plan.

3.3 HCP-ASP

HCP-ASP [55] is a parallel offline algorithm that calls the ASP solver Clingo to compute the branches. The hybrid conditional planner based on actuation actions and sensing actions are represented in answer set programming (ASP) as described in [55]. Feasibility checks are embedded into these action descriptions by external atoms.

3.3.1 Sensing Actions

For instance, occurrences, non-occurrences of sensing actions are modeled by the following choice rule:

$$\{\textit{sense}(\textit{at}(o), t)\}$$

The nondeterministic effects of sensing actions can be expressed using "choice expressions" and "cardinality constraints" as follows:

$$1\{\textit{at}(o, l, t+1) : loc(l)\}1 \leftarrow \textit{sense}(\textit{at}(o), t)$$

This rule describes that if sensing is applied to check the location of an object o (i.e., $\textit{sense}(\textit{atObj}(o), t)$), then we know that the object o is at one of the possible locations l; here, the location l is nondeterministically chosen by the ASP solver.

Chapter 4

4 Assembly Planning

In assembly planning, we are given a set of assembly parts and their initial configurations, and some desired conditions describing the final assembled product. The goal is to find a sequence of manipulation actions that describe which assembly parts are combined in which order to obtain the final product. We propose to solve assembly planning using HCP-ASP. Let us describe our method by the following running example where a table is assembled by a Baxter robot.

4.1 Table Assembly Planning

For instance, consider the assembly of a table, which consists of a top, four legs and four feet as shown in the Figure 3. Initially, the Baxter robot is given a set of legs of varying lengths (e.g., short, tall) and a set of feet of different shapes (i.e., square,triangle,circle) on a bench. The feet can be attached to the legs if the shape of the feet match the hole in the legs. Unfortunately, the robot has partial knowledge about the shapes of the feet, and the connection types of the legs. The robot has to decide for a final configuration that precisely describes the desired product (i.e., which legs are assembled to the table top such that the table is stable, and which feet are connected to those legs), and to generate a plan of actions to reach the final configuration considering all the contingencies.

It is assumed that no tools are required for connecting one part to another, and the orientations of the parts are fixed on the bench. It is also assumed that the Baxter robot can join two parts only when both of them are in hand. We assume here that while joining the two parts, one hand approaches the other to minimize the position error.

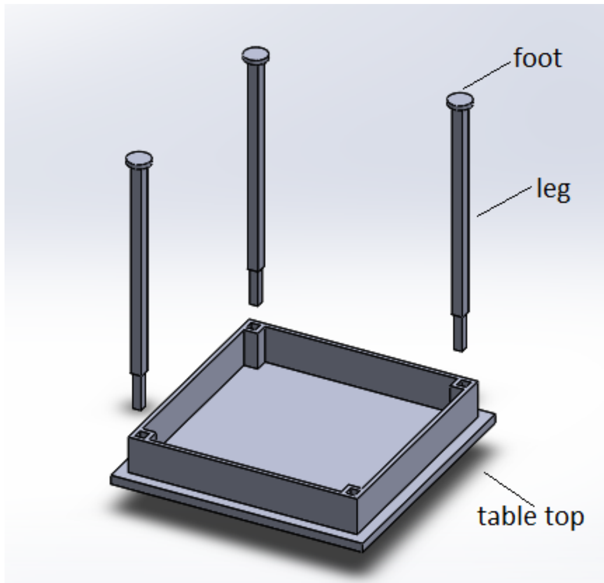


Figure 2: A table assembly problem

4.2 Formalizing Assembly Domain in ASP

4.2.1 Fluents and initial states

In an assembly domain, world states are described by fluents (i.e., atoms whose value change by time). Some of these fluents are fully observable (i.e., the robot knows their values), and some are partially observable (i.e., the robot may not know their values). In the table assembly domain, we consider the following fully observable fluents: *attached*(p, p', c, t), which represents that part p is attached to part p' at attach point c at time step t, and *holding*(m, p, t), which represents that manipulator m of the robot is holding part p at time step t. We consider the following partially observable fluents whose values are identified by sensing action when needed: *shape*(p, s, t), which describes the shape of part p is s at time step t, and $\mathit{color}(p, c, t)$, which describes the color of part p is c at time step t.

Initially, unless told otherwise, we assume that the parts are not attached to each other. This assumption is formalized using defaults as follows:

$$\neg \mathit{attached}(p, p', c, 0) \leftarrow \mathit{not}\ \mathit{attached}(p, p', c, 0).$$

Also, we assume that if a manipulator is free then it is not holding any part p:

$$\neg \mathit{holding}(m, p, 0) \leftarrow \mathit{free}(m, 0).$$

where $free(m, t)$ is a projection of $holding(m, p, t)$.

4.2.2 Actuation actions

In the table assembly domain, three types of elementary actuation actions are considered: $\mathit{hold}(m, p, t)$, which represents that robot manipulator m holds part p at time step t, $\mathit{attach}(m, p', c, t)$, which represents that the robot manipulator m attaches the part p it is currently holding, to another part p' at the attach point c at time step t, and $\mathit{release}(m, t)$, which represents that manipulator m is releasing the part in hand at time step t. An actuation action effects the fully observable fluents directly.

We describe the direct effects of these actuation actions by ASP rules. Consider, for instance, the robot's action of holding the assembly part p at time step $t-1$. As a direct effect of this action, the part p will be in robot's hand at the next time step t:

$$\mathit{holding}(m, p, t) \leftarrow \mathit{hold}(m, p, t{-}1).$$

Similarly, as a direct effect, $attach$ action will join part p in the robot's hand to a part p' at the attach point c,

$$\mathit{attached}(p, p', c, t) \leftarrow \mathit{attach}(m, p', c, t{-}1), \mathit{holding}(m, p, t{-}1).$$

and the $release$ action will cause the manipulator m to be free:

$$\textit{free}(m,t) \leftarrow \textit{release}(m,t-1).$$

We describe the preconditions of actions by constraints. For instance, a manipulator cannot hold a part p, if the manipulator is not free:

$$\leftarrow \textit{hold}(m,p,t), \textit{not free}(m,t).$$

A manipulator m cannot attach a part p, if it is not already holding some other part p'. In this case, it does not matter that the robot attaches to which connection point c so we can just omit c by projecting *attach*(m,p,c,t) predicate to *attachPRT*(m,p,t) as follows:

$$\textit{attachPRT}(m,p,t) \leftarrow \textit{attach}(m,p,c,t).$$
$$\leftarrow \textit{attachPRT}(m,p,t), \{\textit{holding}(m,p',t) : \textit{parts}(p'), p \neq p'\}0.$$

A manipulator cannot join a part p' to part p if the shape of p' is unknown:

$$\leftarrow \textit{attachPRT}(m,p,t), \textit{holding}(m,p',t), \{\textit{shape}(p',s',t) :$$
$$\textit{shapes}(s')\}0, iitype(\textit{Shaped},p').$$

4.2.3 Sensing actions

In the table assembly domain, the robot may not know about the shape/color of the assembly parts, and thus may need to explore the part's shape/color by sensing actions. Sensing actions have nondeterministic effects. For instance, the robot senses the shape of a foot, it can find out one of the regular shapes: circle, triangle, square. This nondeterministic effect is described by a rule as follows:

$$1\{\textit{sensed}(\textit{shapeOfPart}(p,s),t) : \textit{shapes}(s)\}1 \leftarrow$$
$$\textit{sense}(\textit{shape}(p),t-1), \textit{type}(\textit{Shaped},p).$$

Sensing actions can be performed at any time yet there are some necessary domain-dependent conditions which have to be fulfilled to allow the execution of a sensing action. For example, the first condition for sensing the shape of a part p would be that the shape should be unknown. If the robot already knows the shape, it does not need to do unnecessary sensing. This precondition can be expressed by a constraint as follows:

$$\leftarrow \textit{sense}(\textit{shape}(p), t), \textit{shapeOfPeg}(p, s, t), \textit{type}(iiShaped, p).$$

Another precondition should be that robot must be holding the part by one of its manipulators to sense it.

$$\leftarrow \textit{sense}(shape(p), t), \{\textit{holding}(m, p, t) : \textit{manip}(m)\}0, \textit{type}(\textit{Shaped}, p).$$

The outcome of sensing action will provide the missing knowledge about the shape of the part:

$$\textit{shapeOfPart}(p, s, t) \leftarrow \textit{sensed}(\textit{shapeOfPart}(p, s), t), \textit{type}(\textit{Shaped}, p).$$

4.2.4 Concurrency of actions

The table assembly domain allows true concurrency of two actuation actions with different manipulators at the same time step. However, it is not possible to execute two actuation actions by the same manipulator:

$$\begin{gathered} \leftarrow attachM(m, t), holdM(m, t). \\ attachM(m, t) \leftarrow attach(m, p, c, t). \\ holdM(m, t) \leftarrow hold(m, p, t). \end{gathered}$$

Also actuation and sensing actions cannot be performed at the same time:

$$\begin{gathered} \leftarrow \textit{act\_action}(t), \textit{sense\_action}(t). \\ \textit{act\_action}(t) \leftarrow \textit{join}(m, p, c, t). \\ \textit{sense\_action}(t) \leftarrow \textit{sense}(\textit{shape}(p), t). \end{gathered}$$

Similarly, two sensing actions cannot be performed at the same time:

$$\leftarrow 2\{\textit{sense}(\textit{shape}(p), t) : \textit{parts}(P)\}.$$

4.2.5 Existence and Uniqueness constraints

Some state constraints are also required for checking the validity of all the states including the initial and the goal state. For example, every part should have some location (i.e., on the bench or inhand) at all times. The existence of a location is formulated as follows:

$$\leftarrow \{\mathit{loc}(p, r, t) : \mathit{regions}(r)\}0.$$

Similarly, a state will be invalid if a part is located in more than one location. That is why, we also need a uniqueness constraint to correctly model the real world:

$$\leftarrow 2\{\mathit{loc}(p, r, t) : \mathit{regions}(r)\}.$$

Also, there can only be one part attached to another part at the same attach point.

$$\leftarrow 2\{\mathit{attached}(p, p', c, t) : parts(p)\}.$$

4.3 Embedding Commonsense Knowledge in Action Descriptions

In the table assembly domain, the feet can be attached to the legs with similar shaped holes. This commonsense knowledge is embedded in the precondition of $attach(m, p, c, t)$ as follows:

$$\leftarrow \mathit{attach}(m, p, c, t), \mathit{holding}(m, p', t), \mathit{type}(\mathit{Shaped}, p'), \mathit{shape}(p', s', t), 0 = \\ @\mathit{attach\_feasible}(s', p, c).$$

Where $@\mathit{attach\_feasible}(s', p, c)$ is an external atom that checks whether the shape s' of p' matches the hole in part p at the attachment point c. It is commonsense knowledge that a table is stable if it has legs of the same height. This is expressed as a state constraint as follows:

$$\leftarrow \mathit{attach}(m, p, c, t), \mathit{holding}(m, p', t), \mathit{class}(\mathit{Leg}, p'), \mathit{attached}(p'', p, c', t), 0 = \\ @\mathit{check\_stable}(p', p'').$$

4.4 Embedding Feasibility Checks in Action Descriptions

In the table assembly domain, the robot can hold a part if there exists a kinematic solution to reach the part with its manipulator. Such a reachability check can be embedded in the precondition of $hold$ actions as follows:

$$\leftarrow hold(m, p, t), loc(p, r, t), 0 = \&\textit{reachable}[m, p, r, t]().$$

Similarly, the reachability check is needed for the feasibility of $attach$ actions:

$$\leftarrow attach(m, p, c, t), loc(p, r, t), 0 = \&\textit{reachable}[m, p, r, t]().$$

In these constraints, the reachability check is performed by the external atom $\&\textit{reachable}[m, p, r, t]()$, which calls a bi-directional RRT* motion planner [27] from OMPL [47] library through a python file to check for the forward kinematics solution to reach part p with the manipulator m at time t. Such an external python file will return true if there exists a collision-free trajectory to reach part p and false otherwise.

Note that the a task plan is calculated at region level. However, for low-level feasibility checks, exact positions are required. In an attempt to overcome this; we assume that parts are placed in the center of that region. This means if action says that pick part *legr_0* and *legr_0* located in region 1, then the continuous trajectory is calculated from the end-effector position to the center of region 1.

4.5 Planning Problem Description

4.5.1 Initial State

The initial state of a table assembly planning instance is described by a set of facts: $\textit{loc}(p, r, 0)$ represents initial location of each part, $\textit{init\_conn}(p', p'', c)$ represents initial attachment of any two parts p', p" at a connection point c if they are already assembled initially, $\textit{goal\_assembly}(cl_1, cl_2, cl_3, .., cl_m)$ represents the type of parts desired in the table assembly where m is the total number of parts and, $\textit{goal\_conn}(cl_1, cl_2, c)$ represents that the part of class cl_1 should be joined to a part of class cl_1 at connection point c. Additionally, the number of parts required for the final product *numOfParts* is provided as a fact too.

It is important to notice here, with the help of the above mentioned facts, we only provide guidance to the planner, by describing which types of parts should be in the final assembly and joined at which connection point, but we do not specify the part exactly.

A sample goal condition can be that all parts required for the assembly should be assembled in the final product.

$$\begin{gathered} achieved(p,t) \leftarrow attached(p,p',c,t), \\ \textbf{\textit{goal\_conn}}(cl_1,cl_2,c), \textbf{\textit{class}}(cl_1,p), \\ \textbf{\textit{class}}(cl_2,p'). \end{gathered}$$

A part of goal is achieved if p is connected to the p' as goal requires. If all of the parts are attached in this way then our goal has fully acquired.

$$goal(t) \leftarrow numOfParts\{achieved(p,t) : parts(p)\}numOfParts.$$

In incremental model we query until the goal has achieved.

$$\leftarrow query(t), \textbf{\textit{not}}\ goal(t).$$

4.6 Case Study

To demonstrate the assembly planning under uncertainty, we consider the assembly of a table with a top, four legs, and four feet. The problem is to assemble all parts that are required to construct a table as shown in the Figure 3. Initially some parts may be placed on the work bench while some others may already be connected to each other. In this problem, we consider there can be several types of parts on the table (more than the required number of parts). Legs can have varying lengths and connection types, while feet can have different shapes such as square, triangle, circle, of different sizes. The feet can be attached to the legs at their matching holes. The goal here is to decide what the final configuration is (i.e., which legs are assembled to the table such that the table is stable and which feet can be connected to those legs, as they have different shapes and colors), and to generate a sequence of actions to reach that goal assembly configuration.

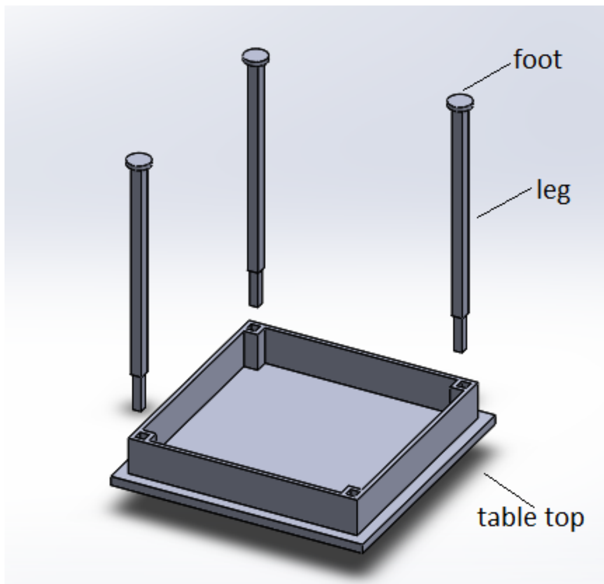


Figure 3: An IKEA furniture table assembly problem

Following are the assumptions of our problem: no tools are required for connecting one part to another, which makes it easy to disassemble parts if they are initially connected to a wrong part. Also, the orientations of parts are assumed to be fixed. We suppose that the connection type of a part is represented by colored marks on the connection ports. The goal is to pick and attach parts in a manner that we can construct the desired assembly. Here it is important to note that by providing the goal conditions but not the exact final state, we are guiding the robot to find a possible assembly configuration, instead of providing a complete goal configuration.

To execute the computed plan as represented in Figure 4, we use Baxter robot with two manipulators. The join action can be performed with a constraint: two parts can be joined only when both of them are in hand. We assume here that while joining the two parts one hand comes towards the other to minimize the position errors.

Assembly planning problem has the following challenges: We need some commonsense knowledge and geometric checks to determine if the attach action between two parts is possible or not, as we do not know the exact final configuration. In motion planning, a collision-free continuous trajectory is hard to find through narrow passages as all parts are very close to each other while performing assembly actions.

4.7 Experimental Evaluations

To evaluate the computational results, we experimented several scenarios with different number of parts, different part types given as an input to construct a table assembly scenarios. Table 1 presents results computed using Hybrid Conditional Planner HCP-ASP [55], and RRT* motion planner [27] from OMPL [47] for the reachability checks embedded into action descriptions. All experiments are performed on a Linux server with 12 2.4 GHz Intel E5-2665 CPU cores and 64GB memory.

In Table 1, to evaluate the scalability of the assembly planning problem, we increased the number of unassembled parts in each problem instance. In instance 1, there are four equal sized legs, three unassembled feet, one already assembled foot, and a table top placed on the work bench. In instance 2, five unassembled legs with different lengths and colors, four feet, and a table top are placed on the work bench while one leg is already assembled to the table top. Instance 3 consists of nine legs, five feet and a table top. For every instance, we increase the number of parts to observe sufficient change in the size of tree. Instance 4 consists of ten legs, five feet and a table top. In instances 3 & 4, the legs can have various lengths, shapes and colors while feet can only have different shapes.

Table 1: Experimental evaluations of the assembly planning scenarios with different number of parts and connection points of the table.

Instance	No. of parts	L(A+S)	DN	BF	N	CPU. Time (in sec)
1	9	12(18+9)	32	3	126	67.124
2	11	14(21+10)	55	3	221	82.325
3	14	21(22+14)	87	3	377	90.645
4	16	34(25+12)	143	3	487	123.098

4.7.1 Results and Discussion

We analyze the quality of a hybrid conditional plan on the basis of the size of tree: the total number L of leaves, the maximum length D of a branch from the root to a leaf, and the number A of actuation, S and of sensing actions in that branch, the total number DN of decision nodes that denote sensing actions in the assembly planning problem, the maximum branching factor BF, the total number N of nodes in the tree. The results of experiments with the above described evaluation metrics are shown in Table 1.

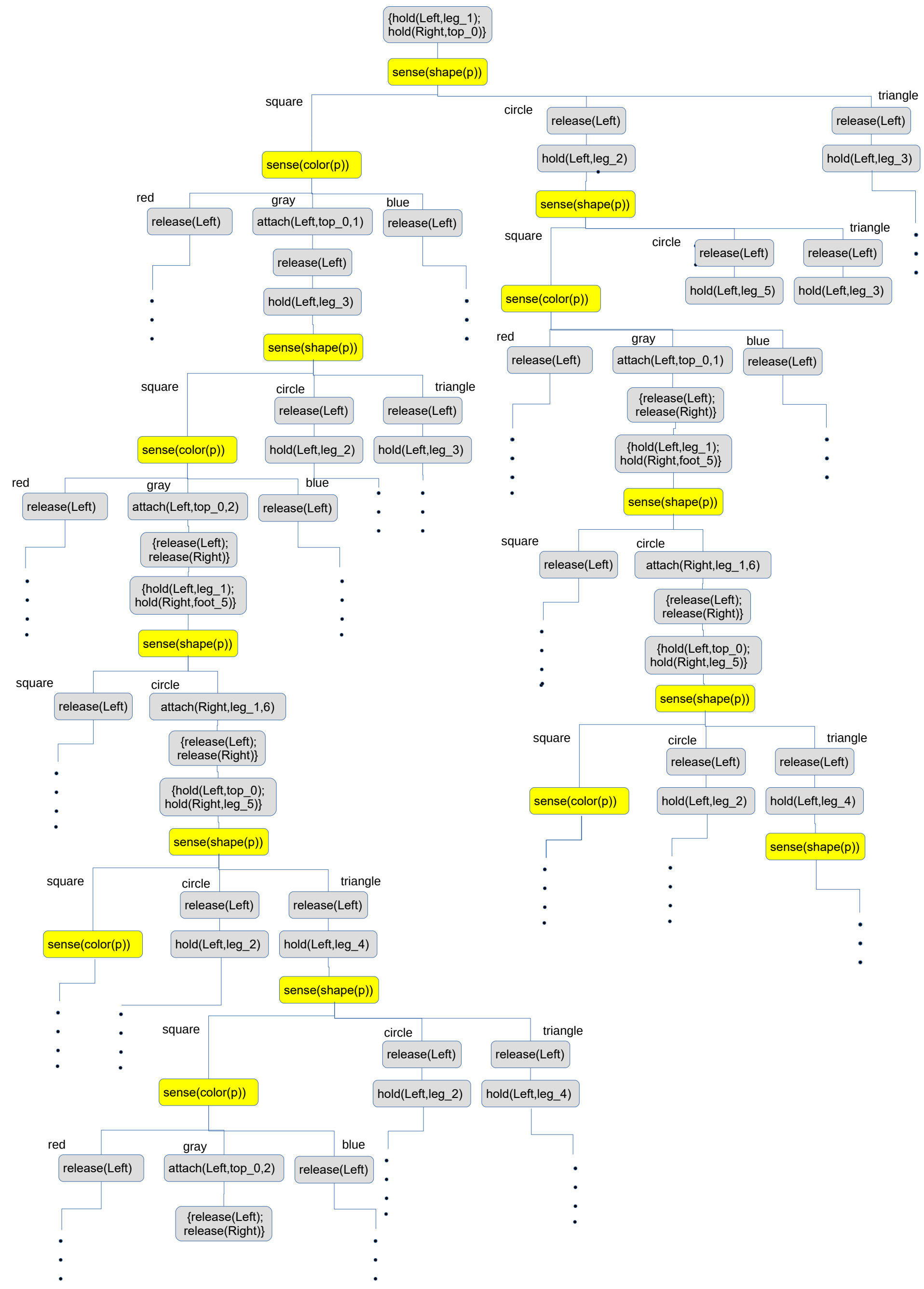
Several trends can be observed here: (i) In instance 1, when no extra parts are given, the total number of nodes in the tree are almost 50% less as compared to the total number of nodes in Instance 2, even if one leg is already assembled in instance 2. (ii) The computation time of a hybrid conditional plan increases as its size increases. For Instance Scenario 4 having 16 assembly parts, a hybrid conditional plan (that consists of 487 actions in total, and 34 different hybrid sequential plans) is computed in about 2.05 minutes. There is a reasonable increase in computation time as the number of sensing actions increases since, the possible outcomes of sensing actions is limited to three different shapes. (iii) The planning time is expected to increase as the degree of uncertainty in the initial state increases since the size of the generated plan will increase. This is what we generally observe, but in instances where the degree of uncertainty in the initial state is the same, it does not take similar planning time to generate the plans.

4.8 Hybrid conditional plan

A hybrid conditional assembly plan for a scenario, where one leg and three feet are already assembled, is given in Figure 4. Initially, seven legs, five feet and a table top are placed on the work bench. Actuation actions are represented as gray colored nodes with only one outgoing edge as an actuation action has a deterministic effect. While, sensing actions are represented as yellow nodes such that each outgoing edge is a possible outcome of that action. Each sensing action has more than one outgoing edge use to its non-deterministic effect. For instance, we have partial information that the color of a leg can be red, blue or gray, so $\mathit{sense}(\mathit{shape}(p))$ has three labeled outgoing edges.

The hybrid conditional plan starts and ends with an actuation action as the robot cannot sense the shape/color of the parts if it is not holding that part. Since each branch of a conditional plan depicts a possible execution of actuation/sensing actions to reach a goal, it is essential that these actions are checked against relevant feasibility constraints.

Figure 4: A hybrid conditional assembly plan



Chapter 5

5 Collaborative Assembly Planning

5.1 Problem Description

Collaborative assembly problem extends the assembly planning problem to accommodate the uncertainties not only in the presence of dynamic environment but also due to the presence of human. The input and output is similar to the one described in the assembly problem Chapter 4. Now as the task is collaborative, the robot has to resolve the uncertainty about human actions and needs to consider those actions while changing his plans accordingly. Furthermore, the communication between human and robot is essential for the interaction to be socially appropriate and safe.

5.2 Approach

As a novel contribution, we propose to extend HCP [55] to include commonsense reasoning and a richer set of communication actions to deal with uncertainty caused by incomplete knowledge about the humans' beliefs, desires, intentions and goals. We consider the following communication actions for collaborative tasks:

(i) robot asking for confirmation (e.g., whether the human will assemble the part she is holding),

(ii) requesting the human to perform some action (e.g., human to unhold a part),

(iii) asking for help (e.g., human to assemble a part that is not reachable to the robot),

(iv) offering help (e.g., when the assembly part is too heavy or needs precision, or when the task is too tedious for the human), and

(v) initiating/ending a conversation (e.g., greeting/acknowledging) and providing explanations.

We propose to model each of these actions formally depending on its type. For instance, communication actions of types (i), (iii) and (iv) require some answers/feedback from humans, and thus they are modeled as nondeterministic actions. These actions serve as decision nodes in a hybrid conditional plan, similar to sensing actions. Requesting human to perform some action, initiating/ending conversations and providing explanations are formalized as deterministic actions. All communication actions have relevant preconditions to ensure that they are executed when appropriate. Furthermore, formalizations of preconditions and effects of communications actions take into account commonsense knowledge for more natural communications.

With such a general HCP framework, with sensing actions, robots can identify which assembly part the human workmate is holding; with commonsense reasoning, robots can conclude that a heavy assembly part cannot be moved by a human; and with communication skills, robots can communicate with humans in different ways for safer and effective collaborations. To the best of authors knowledge, HCP has not been used for collaborative assembly planning.

5.3 Extension of Fluents

Partially observable fluents are extended to represent the sensed human actions and mental state(intentions and desires). Inertia is not defined on such partially observable fluents as they are changing instantly. To represent the belief state of the robot about human behavior, the fluents are defined: *holdingH*(t) (human is holding something at step t); *holdingHPart*(p, t)(human is holding part p at time step t); *attachedH*(p, p', c, t)(human attached part p to p' at connection c at time step t). Additionally, to represent the robot's belief state about human intentions and desires, the following fluents are designated:

(i) *wantToAttach*(p, p', t) (human intention to attach p to p')

(ii) *acceptToAttach*(p, p', t) (human willingness to attach p to p' when the robot asks for help)

(iii) *acceptRobotToJoin*(p, p', t) (human willingness to accept the help offered by the robot to attach p to p')

5.4 Extended Modeling of Sensing Actions

Sensing actions considered in this problem are as follows:

(i) *sense*$(humanHolding, t)$ (sensing if human is holding anything or not at time step t)

(ii) *sense*$(humanHoldingWhichPart, t)$ (sensing that human is holding which part at time step t)

(iii) *sense*$(humanUnholding(p), t)$ (sensing if human is unholding part p at time step t)

(iv) *sense*$(humanAttachingWhere(p, p'), t)$ (sensing where is human attaching the parts p and p' at time step t)

Effects For collaborative assembly problem, the effects of sensing actions are defined over the belief states of partially observable fluents about human actions. Depending on the type of possible outputs of action, a sensing action can have a boolean or a multivalued output. A boolean fluent should have a negative correspondent while multivalued fluent does not need a negative correspondent. Sensing actions must only effect external predicates. Here external predicates are denoted a keyword $sensed$ which can be seen below. For instance, if the robot senses whether human is holding something then the output of such action will be boolean i.e. the human is holding something or not. Such an effect can be demonstrated as:

$$1\{\textit{\textbf{sensed}}(holdingH, t); sensed(notholdingH, t)\}1 \leftarrow \\ \textit{\textbf{sense}}(humanHolding, t-1).$$

Similarly, sensing whether the robot is releasing a part p. The robot may be holding or not holding p.

$$1\{\textit{\textbf{sensed}}(holdingHP(p), t); \textit{\textbf{sensed}}(\neg \textit{\textbf{holdingHP}}(p), t)\}1 \leftarrow \\ \textit{\textbf{sense}}(humanUnholding(p), t-1).$$

On the other hand, the output of the following sensing action can range through all the available parts. Note that, for simplicity this is an assumed here that human can only hold one part at a time.

$$1\{\textit{\textbf{sensed}}(holdingHPart(p), t) : parts(p)\}1 \leftarrow \\ \textit{\textbf{sense}}(humanHoldingWhichPart, t-1).$$

While we sense "in which attach point the human is attaching the assembly part p" the sensed output can be any of the free connection points available on the part p'.

$$1\{\textit{\textbf{sensed}}(attachedH(p, p'c'), t) : connPoints(c')\}1 \leftarrow \\ \textit{\textbf{sense}}(humanAttachingWhere(p, p'), t-1).$$

Preconditions As discussed in Chapter 4, to be able to execute a sensing action some domain-dependent conditions need to be met. One of the preconditions of sensing the part which human is holding is the robot should be holding something at that time. Here two sensing actions are considered first, to check if human is holding something and second, the robot can sense the part human is holding. If we directly sense the part then the tree branching factor will increase and the plan tree get more computationally expensive.

$$\leftarrow \textit{sense}(humanHoldingWhichPart, t), \textit{\textbf{not}}\ holdingH(t).$$

Along the same lines, a robot can not sense attach action if the human is not requested or asked to attach.

$$\leftarrow \textit{sense}(humanAttachingWhere(p, p'), t), \textit{\textbf{not}}\ \textit{requestedAttach}(p, p', t).$$

5.5 Modeling Communication Actions

In addition to actuation actions and sensing actions, we also consider communication actions:

(i) *confirmAttach*$(p, p'), t)$ (confirming if human wants to attach p to p' at time step t)

(ii) *askHelp*(p, p', t) (asking human help in attaching part p to p' at time step t)

(iii) *offerHelp*(p, p', t) (offering help in attaching part p to p' at time step t)

(iv) *requestUnhold*(p, t) (requesting human to un-hold part p at time step t)

(v) *requestAttach*(p, p', t) (requesting human to attach part p to part p' at time step t)

5.5.1 Communication Actions having Deterministic Effect

Requesting a human to perform some action, initiating/ending conversations and providing explanations are formalized as deterministic actions, like actuation actions.

The nondeterministic communication actions serve as decision nodes in a hybrid conditional plan, similar to sensing actions. The formalizations of the preconditions and effects of communications actions take into account commonsense knowledge.

$$\textit{requestedAttach}(p, p', t) \leftarrow \textit{requestToAttach}(p, p', t-1).$$

5.5.2 Communication Actions having Non-Deterministic Effect

The communication actions (e.g., asking for confirmation) that require some answers/feedback from humans are modeled as nondeterministic actions, like sensing actions. For instance, after the robot asks human help in attaching part p' to p, when it is unable to reach p, in return the human responds affirmatively or negatively:

$$\begin{aligned}&1\{\textit{acceptToAttach}(p, p', t);\\ &\quad \neg\textit{acceptToAttach}(p, p', t)\}1 \leftarrow\\ &\quad \textit{askHumanHelpAttach}(p, p', t-1).\end{aligned}$$

Preconditions All communication actions have relevant preconditions to ensure that they are executed when the appropriate conditions hold. For instance, the robot can ask the human teammate for help in assembling part p to p' if it really cannot reach the part p with any of its manipulators:

$$\begin{aligned}&\leftarrow \textit{askHelp}(p, p', t),\\ &\quad \textit{not}\ 2\{\textit{reachabilityFails}(m, p, t) : \textit{manipulator}(m)\}.(1)\end{aligned}$$

and if the human is not already holding some other part p'':

$$\begin{aligned}&\leftarrow \textit{askHelp}(p, p', t),\\ &\quad \textit{humanHoldsPart}(p'', t) \quad (p \neq p'').\end{aligned}$$

Here, the reachability check is embedded into the definition of *reachabilityFails*(m, p, t) by an external atom, in the spirit of [12]. Continuous reachability check is performed using RRT* motion planner [27] from the OMPL library [48].

A precondition for confirm attach communication action is that the robot can not ask if human wants to help joining two parts which are not designed to be attached.

$$\begin{aligned} \leftarrow\ & \textit{confirmAttach}(p, p', t), \\ & \textit{class}(cl, p), \textit{class}(cl', p'), \\ & \quad \textit{not attach\_relation}(cl, cl', \_). \end{aligned}$$

Confirm communication action can only be performed only if human wants to help by showing a sign by holding that part.

$$\leftarrow \textit{confirmAttach}(p, p', t), \textit{not holdingHPart}(p, t).$$

5.5.3 Ramifications

As we already discussed that the outcome of sensing action will indirectly effect the knowledge on matter by a partially observable fluent.

$$\textit{attachedH}(p, p', c', t) \leftarrow \textit{sensed}(attachedH(p, p'c'), t).$$

An important ramification occurs when human attaches a part, his actions will effect the attached fluent (fully observable). These two fluents should be linked to keep track of the progress made by both robot and human.

$$\textit{attached}(p, p', c', t) \leftarrow \textit{attachedH}(p, p', c', t).$$

5.6 Integrating Commonsense Knowledge

Since the feet of the table has a sharp nail and is dangerous for human to assemble feet to the legs, the robot offers help to the human for such tedious and unsafe tasks. For safety concern, robot should not allow human to attach the part which is dangerous for human to join.

$$\leftarrow \textit{sense}(confirmAttach(p, p'), t), type(Dangerous, p).$$

If the human is holding a part that can be assembled to the part that the robot is holding, then (instead of trying to pick it from the human) he needs to confirm with the human as to whether she will assemble the part. If the human's response is negative, then the robot requests the human to unhold the part.

$$\begin{aligned}1\{&\textit{wantToAttach}(p, p', t);\\ &\neg\textit{wantToAttach}(p, p', t+)\}1 \leftarrow\\ &\quad\textit{confirmAttach}(p, p', t).\end{aligned}$$

5.7 Feasibility Check Integration

In the assembly planning problem, feasibility checks are added as hard constraints as the robot is physically not capable to perform such actions. However, in collaborative assembly problem, the robot can resolve its inability to perform an action by asking for help from the human team-mate when the robot fails to perform a task. To enable communication for such cases, we do not add reachability check failure as a hard constraint, but include it as a weak constraint to the domain. We want to penalize such failures as much as possible and if we can not avoid them, then these cases act as a precondition for the communication actions, where robot asks human help. The relevant ASP rule read as:

$$\textit{reachableFail}(m, p) \leftarrow \textit{hold}(m, p, t), \textit{loc}(p, r, t), 0 = @reachable(m, r).$$

Weak constraint defined below penalizes a solution whenever a reachability check fails but still provides the best possible solution. In this rule, 2 is the weight of how much a failure should penalize and 1 is the priority, as we also have weak constraints to avoid extra actions with lower preference.

$$:\sim \textit{reachableFail}(m,p).[2@1]$$

5.8 Human Preferences

We embed verbosity and safety levels as human preferences. In our proposed framework, communication is initiated when needed, avoiding e.g., unnecessary calls for human help. Depending on the verbosity level, the robot can provide explanations. For instance, if the verbosity level is high the robot can tell that it cannot reach the assembly part as an explanation for its request for help. Depending on the safety level, the robot can offer help. For instance, if the safety level is high, the robot can offer help when the human attempts to pick a sharp object.

$$\leftarrow \textit{offerHelp}(p,p',t), \textit{type}(Dangerous,p), \textit{not safety\_level}(2).$$

Chapter 6

6 Case Study – Collaborative Assembly Planning

We have considered instances of furniture assembly planning, that include different types of collaboration scenarios:

(S1) If the robot senses that the human is holding a part that can be attached to what the robot is holding, then the robot confirms with the human about her intention of attaching the parts and safely allows her to attach the parts.

(S2) If an assembly part is not reachable by the robot and he senses that the human is free, then the robot asks for help in attaching that part to what he is holding.

(S3) If the robot senses that human is holding a part which is tedious to attach, then he offers help in attaching parts.

A conditional plan shown in Figure 5 computed for one of these instances, and the dynamic simulation of execution of this plan (prepared in Gazebo with ROS interface) are shown in the video at `http://cogrobo.sabanciuniv.edu/demos/hri/HCP_HRI_demo_video.mp4`. Snapshots of this simulation are shown in Figures 6 and 7.

In Figure 6, we see that the robot first assembles one of the legs to the table top. Then he notices that the human is holding a table leg, which can be assembled to the table top. The robot confirms with the human as to whether she is planning to attach the leg to the table top. Notice that, the robot asks for confirmation only when the human wants to collaborate. If the human is performing some other task

Table 2: Human-Robot Interaction

Scenario 1	
Type of Interaction	**Confirmation**
Observation	Robot senses that human is holding part P1.
Confirm	Robot: Do you want to join P1 to P2? Human: Yes, I want to join P1 to P2.
Request	Robot: Please join.
Observation	Robot senses human's joining action.
Scenario 2	
Type of Interaction	**Asking for Help**
Observation	Robot senses that human is not holding anything. (Checking availability)
Ask help	Robot: Can you help me in joining part P to P1? Human: Yes, sure.
Observation	Robot senses human's joining action.
Scenario 3	
Type of Interaction	**Offering Help**
Observation	Robot senses that human is holding a heavy part P.
Offer help	Robot: Can I help you in joining part P. Human: Yes. Robot: Please, unhold part P so that I can join.
Observation	Robot senses human's unhold action.

which do not concern the robot's assembly task then the robot does not execute a communication action.

In Figure 6, after the human confirms affirmatively that she is planning to attach the leg to the table top, we see that the robot requests the human to attach it. After the human attaches the second leg, the robot assembles the third table leg. Then the robot notices that the last leg he plans to assemble is far from him, so he cannot reach it. Then the robot asks human for help in assembling the last leg. After the human assembles the last leg, the robot acknowledges. It is important to note here, that we consider human to be collaborative too that is why the request actions do not need any response.

6.1 Dynamic Simulation

In order to demonstrate the dynamic simulation of a hybrid conditional plan computed for a collaborative assembly of a table, we use ROS Gazebo simulation environment. We integrate geometric kinematic forward reachability check during the planning phase in order to ensure plan feasibility. The hybrid conditional plan generated by HCP-ASP for an instance of collaborative table assembly scenario is shown in Figure 5.

In the hybrid conditional plan, sensing actions are represented by yellow nodes, actuation actions by gray nodes and communication actions by pink nodes. Communication actions can have one or more outgoing edge depending on the type of communication action.

In the Figure 6, snapshot 1 shows the initial state of the work bench setting, where legs of varying shapes, sizes and colors are placed on the work bench. In snapshot 2, the robot holds one leg, and attaches it to the table top in snapshot 3 & 4. After attaching one leg, in snapshot 5, he notices that human is holding a leg and in snapshot 6, it confirms human intention by asking if she wants to assemble the table leg. The human replies affirmatively that she wants to assemble that leg to the table top.

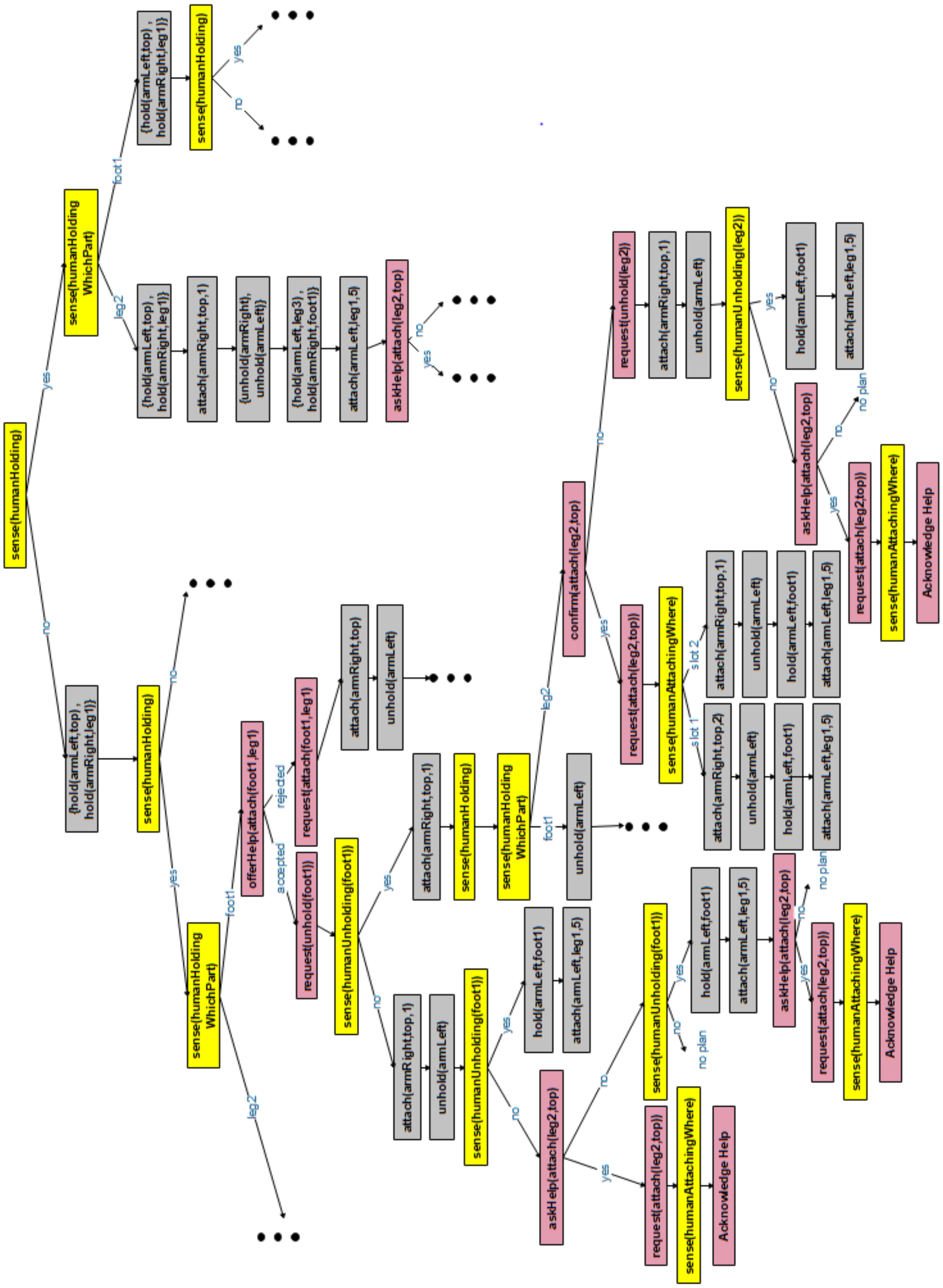


Figure 5: Hybrid Conditional Plan for Collaborative furniture assembly instance

Figure 7 is the continuation of the plan. In snapshot 7, the robot requests her to attach the leg while he senses her attach action. In snapshot 8, the robot continues to execute his plan by assembling another leg. After assembling the leg, he notices that it is unable to attach the last leg as the reachability check failed, so in snapshot 9, it asks human for help. Human accepted to help the robot in assembling that leg, so the robot requests her to assemble the leg in snapshot 10. In snapshot 11, the robot senses that the human has attached the leg and the task is complete so he acknowledges human help by saying "Thank you" in snapshot 12.

Figure 6: Simulation snaphots: Part 1. The robot assembles one of the legs to the table top. Then he notices that the human is holding a table leg, which can be assembled to the table top. The robot confirms with the human as to whether she is planning to attach the leg to the table top.

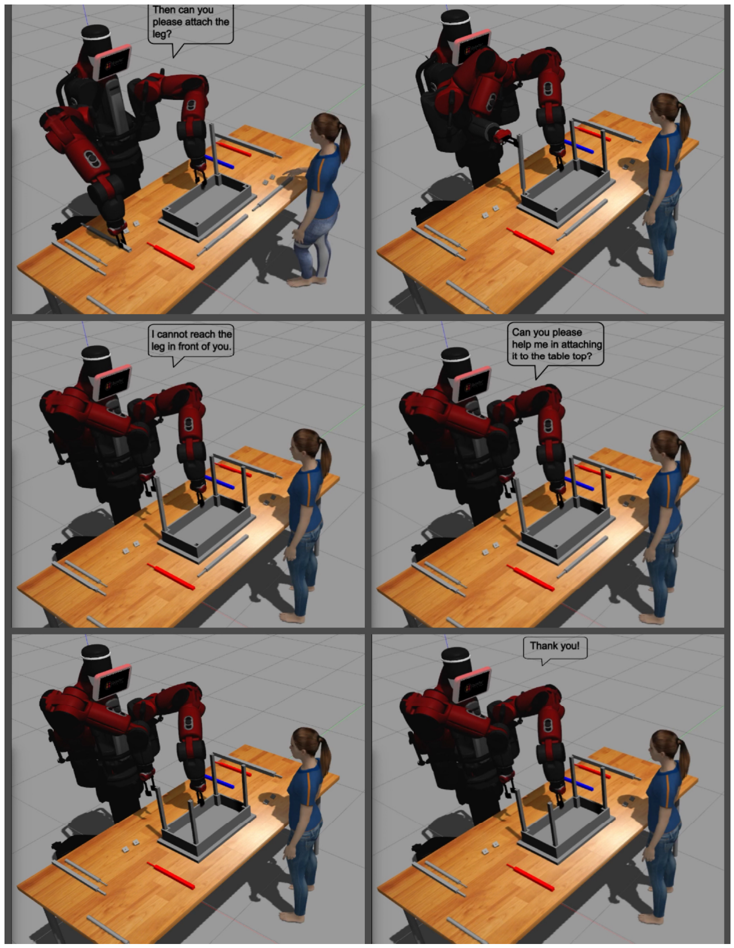


Figure 7: Simulation snaphots: Part 2. After the human confirms affirmatively that she is planning to attach the leg to the table top, the robot requests the human to attach it. After the human attaches the second leg, the robot assembles the third table leg. Then the robot notices that the last leg he plans to assemble is far from him, so he cannot reach it. Then the robot asks human for help in assembling the last leg. After the human assembles the last leg, the robot acknowledges.

6.2 Computational Results

In our experiments, we have used the HCP planner HCPASP [55] for generating conditional plans, and RRT* motion planner [27] from OMPL [47] for the reachability checks embedded into action descriptions. All experiments are performed on a Linux server with 12 2.4GHz Intel E5-2665 CPU cores and 64GB memory.

We have considered instances of furniture assembly planning, that include different types of collaboration scenarios: (S1) If the robot senses that the human is holding a part that can be attached to what the robot is holding, then the robot confirms with the human about her intention of attaching the parts and safely allows her to attach the parts. (S2) If an assembly part is not reachable by the robot and he senses that the human is free, then the robot asks for help in attaching that part to what he is holding. (S3) If the robot senses that human is holding a part which is tedious to attach, then he offers help in attaching parts.

6.2.1 Results and Discussion

We have analyzed the effects of the following objective measures on the computation time: the total number L of leaves, the maximum length D of a branch from the root to a leaf, and the number A of actuation, S of sensing and C of communication actions in that branch, the total number DN of decision nodes that denote sensing actions and nondeterministic communication actions, the maximum branching factor BF, the total number N of nodes in the tree. The results

Scenario	Instance	L (A+S+C)	DN	BF	N	CPU Time (sec)
S1	1	24 (5+9+6)	84	4	172	285
	2	32 (8+12+8)	144	6	201	425
S2	1	20 (4+9+6)	64	4	120	315
	2	17 (5+12+8)	70	4	130	350
S3	1	36 (15+17+12)	201	6	357	750
	2	48 (25+28+16)	325	8	523	1520

Table 3: Experimental evaluations of the three types of collaboration scenarios S1–S3.

of experiments with these objective measures are shown in Table 3.

There are several important observations. (i) The computation time of a hybrid conditional plan increases as its size increases. For Instance S3–2, a hybrid conditional plan (that consists of 523 actions in total, and 48 different hybrid sequential plans with a makespan less than 69) is computed in about 25 minutes. The increase in computation time is not surprising since, even for polynomially bounded plans with limited number of nondeterministic actions, the complexity of conditional planning is Σ_2^P-complete [2]. On the other hand, note that the plan is computed offline considering all possible contingencies, and thus no time is spent for planning during execution. (ii) The average computation time of a branch of the tree, which represents a possible hybrid sequential plan to reach the goal, is the total CPU time divided over L. This suggests that, if a hybrid sequential plan of actuation actions were computed instead of a hybrid conditional plan, then replanning would take around half a minute for Instance S3–2. Such (re)planning times are not acceptable while communicating with a human. Therefore, computing a hybrid conditional plan in advance for collaborative assembly tasks that involve communications is advantageous.

Chapter 7

7 Human Subject Experiments

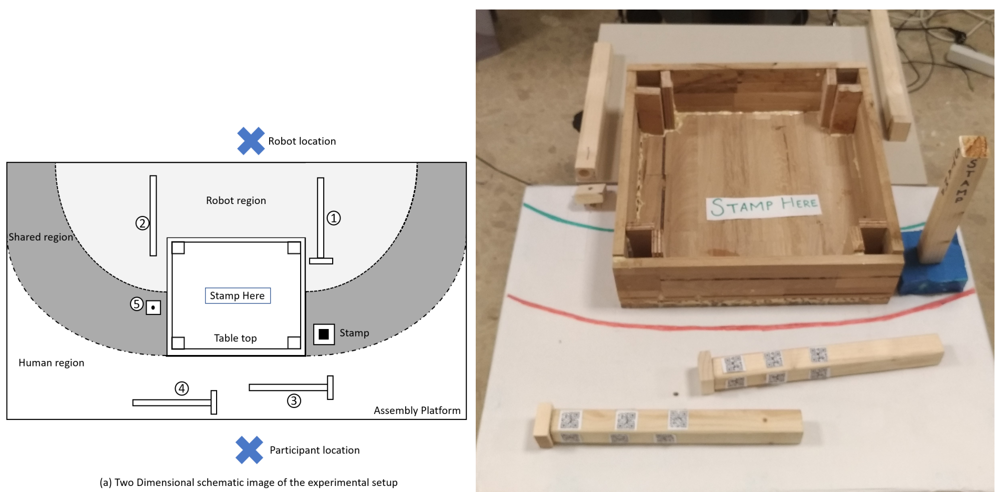
For a more comprehensive evaluation of our HCP-based method for collaborative assembly planning, we devised a physical experiment where a human and robot collaboratively assemble a furniture table. Initially all the assembly parts are provided i.e. a table top, four legs, one unassembled foot of the leg (other three feet are already assembled), and a stamp to label the table with the company logo. Goal of the assembly task (given to both human and robot) is to assemble the furniture table. Since the proposed method is focused on human-robot collaborations, we perform experiments with objective and subjective measures in the spirit of [32, 52], by means of a survey applied to a diverse group of participants.

The experiment is designed such that the human participant can experience four collaboration scenarios: (i) robot asking for help when a feasibility check fails; (ii) robot asking for help when the workplace is too close to human (risk to human safety); (iii) robot offering help to the human team-mate because the task is dangerous; and (iv) robot confirming human intention if he is holding a part.

To compare our method of integrating communication actions at the planning level, we performed two sessions of experiments: one where robot communicates in order to exchange each other's intentions and to provide explanations for it's actions, and second when there is no communication action. These physical experiments are implemented with a Baxter robot, a collaborative robot from Rethink Robotics, which is designed to work effectively alongside people in a factory setting, making it possible to deploy in environments.

The goal of the task is to first stamp the table with the company label and then assemble all the legs and foot to obtain the end product. Assembly parts may have certain objects which are sharp and therefore dangerous for human, or some objects which are placed close to human and it is preferred that robot should not enter that region without prior warning. Also there is accessibility issue for both human and robot. Some parts may not be accessible to the robot. While some parts are not reachable by the robot. To make use of each other's abilities and to overcome each others weaknesses, the robot must plan for actuation, sensing and communication actions. Communication actions may also affect the social understanding and trust of human towards the robot. Therefore, to evaluate such a system, not only objective measures are important but also some abstract qualitative surveys to better understand the emotional state of human during the experiment.

7.1 Experimental Setup



(a) Two Dimensional schematic image of the experimental setup

Figure 8: Experimental Setup: (1) leg1; (2) leg2; (3) leg3; (4) leg4; (5) unassembled foot. Foot is a dangerous object for human to hold as it has a sharp nail attached to it. While, safety levels are also defined based on the regions. Robot can manipulate anything safely in robot region shaded as light gray (safety level 0); in the shared region shaded as dark gray, the robot can not manipulate objects safely without prior communication (safety level 1)

The experiment was organized to demonstrate a furniture assembly task on a platform. The human and robot are standing on the two opposite sides of the table facing each other, as shown in the Figure 9. The platform is divided into three regions; the white area where only robot can have access, the green area where both human and robot can reach(shared region), and the red area where only human can have access. Assembly parts of the furniture desk are placed such that: (i) leg1, leg2 and an unassembled foot of the leg are placed in the region which is accessible to the robot only(safety level 0) (ii) leg3, and leg4 in the region accessible to the human participate only (iii) the company label stamp in the shared region (iv) while the table top in the shared region as shown in Figure 9.

The experimental setup poses some challenges. First, the stamp in the shared region should either be used by the human or robot should reach the stamp with a warning for human to maintain a safe distance. Second, the foot is a sharp object which might hurt human that is why robot need to make sure to avoid such safety hazards. Third, some parts are not reachable by the robot and communication is required to resolve such inability of the robot.

A camera was attached to a stand near human so that it can sense when human is holding an assembly part, exactly which part he/she is holding and in which slot the human is attaching the part. For communication in natural language, we have used a Python Library gTTS (Google Text-to-Speech), a CLI tool to interface with Google Translates text-to-speech API. While, human speech recognition is performed using Google Speech API.

7.2 Participants

We conducted the physical experiment with 25 volunteers (12 female, and 13 male volunteers) with diverse academic backgrounds. The volunteers' age varies between 23 to 36. The volunteers had no prior experience of human robot collaboration. The task of both agents (human and robot) is to assemble the furniture table with the assembly parts accessible to each of them. The roles of agents are not defined prior to the experiment, human and robot decide for the task allocations through communication.

7.3 Experimental Procedure

Each participant experienced a total of three experimental sessions to test the formal framework explained above, using a human-robot collaborative assembly of a coffee table (a tabletop task). Before the experiment, initial instructions were given to all the participants which are kept consistent through all the volunteers. To make the task challenging, participants were not allowed to assemble two legs consecutively.

The first session was performed to train the human with all assembly tasks and to make him aware about what kind of interactions can be experienced. Training was conducted as part of the experimental procedure since the participants have no expertise for such a task.

Our experimental sessions consisted of two randomly selected groups of participants: those who first experienced a human robot collaboration with communication, and those who initially experienced a human robot collaboration without any communication or explanation. That is, each group experienced both experimental conditions but with different order. Breaks were scheduled after each experimental session.

Figure 9: Physical experiment: In snapshot (1) the robot explained human that the stamp is too close to you, it will be safer if she can stamp the table; (2) the robot continues with the next assembly task; (3) the robot senses that human is holding a leg and confirms whether she wants to assemble it; (4) after the human assembled, robot assembles another leg; (5) robot asked human help to assemble a leg as it is not feasible for the robot to reach the leg (6) the robot picks foot with the sharp nail (dangerous task for human) to assemble it to the leg

7.4 Quantitative Evaluation

As the collaborative assembly planning system described previously in Chapter 7 is formalized to plan for communication actions such that they can be executed immediately when needed and provide safer collaborations. For this reason, the quantitative performance measures selected to evaluate the HCP-based system for human robot collaboration are representatives of the following properties: interaction time , task fluency, safety hazards, and task completion. For future references, we will call the trials verbose (when there is communication between human and robot to convey intentions, giving explanations e.t.c.) and non-verbose (when there is no communication between human and robot) experiment.

During the execution, it is observed that the experimental sessions with communication between human and robot were significantly faster. The analysis of mean and variance by using t-test over the calculated interval times show statistically significant differences (p $<$ 0.0131). The **overall interaction time** (in minutes) in the verbally active human-robot collaboration has a distribution ($t_{interaction\_verbose} = 4.12 \pm 0.5 mins$). While the time taken by the experiment without communication was more than that by the verbose one ($t_{interaction\_non-verbose} = 5.58 \pm 1.6 mins$). To further evaluate the reason behind this time difference, we have analyzed the assembly time, and idle time.

The **assembly time** is the total time taken by all the actuation actions performed either by robot or human. The mean time taken by the assembly actions in the verbose session is $t_{assembly\_verbose} = 4.23 \pm 1.87 mins$. The deviation in the assembly time is high because of the different allocation of tasks. Meanwhile, the assembly time distribution in the non-verbose version is comparable to the verbose ($t_{assembly\_non-verbose} = 4.31 \pm 1.34 mins$). The t-test results show that there exist no statistically significant difference. By conventional criteria, this difference is considered to be not statistically significant. The definition of **idle time** considered here, is the duration where both human and robot are not performing any action. We are interested in the idle time since during this time there is no progress towards the goal. The idle time in the non-verbose experiment is

$t_{idle\_non-verbose} = 1.5 \pm 0.3 mins$. Whereas, in verbose experiment the distribution is $t_{idle\_verbose} = 0.4 \pm 0.08 mins$. We can observe that when there is no communication, the increase in idle time is statistically significantly higher (p $<$ 0.0001) due to the confusion of team-mate in understanding each others intentions, more time is spent in sensing rather than actually performing actions.

7.4.1 Safety

When there is no communication between human and robot, subjects reached the sharp object **60**% of the time. Without the robot's warning about the dangerous objects, the participant has no information that the object is dangerous for him. That is why, the safety code is violated by most of the volunteers in the absence of communication. This result indicate the importance of embedding communication actions into the planning.

For the first action, i.e. stamping the table top with the company label, the idle time is observed to be the highest. The reason is that the robot could not express his concern that the human should keep the distance from the stamp while the robot is performing that task. Also the robot does not know if human will stamp or not. This confused situation takes time to resolve and due to this state of uncertainty, **25**% of the participants reached the stamp at the same time as robot reached it. This also caused safety hazard as the robot is unable to explain his situation when there is no communication.

7.4.2 Task Completion

In the experiments, when there was no human-robot communication, if the robot is unable to perform an action it cannot ask help from the team-mate. The participants were not able to understand why the robot is not attaching the last leg and 30% of the times, the participants waited too long for the robot that the task seemed incomplete at the end. The resulting behavior indicates that communication also ensures task completion.

7.5 Qualitative Evaluation Measures

We have also performed two types of surveys: first survey to evaluate if volunteers feel safer when there is communication while carrying out collaborative assembly tasks and if the volunteers like the level of verbosity (i.e., robot acknowledging human actions and explaining his actions to human); second survey that evaluates the human mental load during assembly tasks, provided by NASA task load index.

After the experimental trials, participants were asked to fill these surveys to collect qualitative data regarding the experiment. As every participant evaluates the system subjectively, we used 5 point Likert scale together with some open ended questions to better understand how participants feel around the robot teammate. Likert-scale questions were asked about each type of communication mode, and the performance measures of safety, verbosity and perceived intelligence were measured. The collected Likert responses were then additionally analyzed by the mean and standard deviation across each question. For example in Table 4, Likert responses were asked '1' being the least safe and '5' being the most safe as compared to our experiment without communication. The mean value shows that most of the participants considered it safer when there is communication between human and robot.

To assess the reliability or internal consistency of the collected response scales, i.e. to determine how closely related a set of items are as a group, Cronbachs's alpha measure is employed. For a survey to be acceptable, the threshold of Cronbach's alpha is commonly taken as $\alpha \geq 0.7$. We can observe that all Cronbach's alpha values are above 0.7, indicating that the acquired survey data is consistent and reliable.

In Table 4, a rating of 4.8 for Q1 means that most volunteers felt much safer when the robot confirmed their intentions before doing his task. Similarly, when the robot offered help for the foot which has a sharp nail the volunteers appreciated the fact that robot is concerned for their safety indicated with 4.8 in Q2. Furthermore, when the robot tried to remain at a safe distance from the human in Q3, some participants gave satisfactory responses as they found it excessive. On average, human volunteers felt safer during the whole interaction.

In Table 5, participants acknowledged that verbal communications are helpful in collaborative tasks. They enjoyed the conversation in Q5, when robot explains why he needs help or why the help is being offered by the robot.

Furthermore, to evaluate the overall performance of the general framework and quality of the selected task, NASA task load index was used as provided in Table 6. The NASA task load index (NASA TLX) is a tool for assessing and conducting a subjective mental workload (MWL) assessment. In Table 6, the statistical data shows that 50% of the participants did not consider the task to be mentally or physically demanding. The encouraging fact is that people were neither stressed or annoyed during the interaction as the rating is quite low.

Table 4: The survey questions and their summary statistics: The mean values closer to the maximum Likert-scale value of 5, demonstrate that the participants considered the interaction safer when the robot communicates during the specific scenario.

(Safety) How safer human felt during the interactions(when there is planning with communication)?			
	Cronbach's alpha	Std. Deviation	Mean
Q1. When the robot confirmed about human intention of performing an assembly?	0.75	0.41	4.80
Q2. When the robot offered for help, as you intended to handle a dangerous assembly part?	0.73	0.41	4.80
Q3. When the robot tries to remain at a safe distance from you?	0.75	0.60	4.6
Q4. While collaborating with the robot?	0.75	0.60	4.45

7.6 Results

We have presented the experiment design and implementation of a human-robot collaborative assembly system with and without communication. We have reported evaluation results on a user study in which unexperienced volunteers interact with the robot system and fill out the surveys to rate different aspects of the human satisfaction. Evaluation has been performed on the hypothesis that a hybrid planning with communication actions along with the integration of human

Table 5: The survey questions and their summary statistics: The closer the mean values to the maximum Likert-scale value of 5, the more participants liked that the robot being verbose.

(Verbosity) How much did you like communication during the interactions?			
	Cronbach's alpha	Std. Deviation	Mean
Q5. When the robot confirmed with you before you perform your assembly?	0.71	0.75	4.65
Q6. When the robot provided an explanation (safety) as to why human help is needed?	0.74	0.41	4.80
Q7. When the robot provided an explanation as to why help is offered?	0.72	0.55	4.75
Q8. When the robot provided an explanation (reachability) as to why human help is needed?	0.75	0.57	4.7
Q9. How useful was the communication overall?	0.74	0.74	4.65

Table 6: The survey questions and their summary statistics: The mean values closer to the minimum Likert-scale value of 1, means the participants find the task to be less mentally and physically demanding, the pace of the task less hurried, and the participants were less annoyed by the task. The result also showed that the participants were mostly successful in accomplishing the task. Also the participants considered the collaboration moderately useful in real life.

NASA Task Load Index			
	Cronbach's alpha	Std. Deviation	Mean
Q10. How mentally demanding was the task?	0.79	1.13	2.30
Q11. How physically demanding was the task?	0.79	1.04	2.15
Q12. How hurried or rushed was the pace of the task?	0.74	0.80	2.10
Q13. How successful were you in accomplishing what you were asked to do?	0.7483	0.41039	4.80
Q14. How useful was the collaboration overall?	0.73	1.14	2.45
Q15. How insecure, discouraged, irritated, stressed, and annoyed were you?	0.74	0.57	1.30

preferences can increase task efficiency, human satisfaction, safety, and perceived intelligence.

We analyze the objective measures like time, success rate, task completion and safety. We recorded those interactions to measure the interaction patterns and performed t-test to show the statistical significance in performing with and without communication. The resulting performance measures indicate that collaboration with planned communication actions between human and robot are statistically significantly safer and have better success rate.

Subjective measures have been used to measure factors which cannot be calculated but experienced by human while interacting with the robot. Each user may perceive the interaction differently. Such factors have been measured through the data collected by the questionnaires provided to volunteers after the experimental session, so that the verbose and non-verbose experimental session can be compared. The results of the questionnaire provide evidence that majority of the volunteers found interactions to be safe and enjoyed the verbal communication. Volunteers were mostly impressed by the explanations provided by the robot as this is perceived to indicate robot intelligence.

Chapter 8

8 Conclusion

In the uncertain and human-centric environments, the use of hybrid conditional planning provide encouraging results from the following perspectives. (i) Formal modeling of communication actions, embedded with formal representation of commonsense knowledge and low-level geometric checks, helps the robots to better understand when to communicate and how, as part of planning their actions. (ii) Offline planning of actions considering all contingencies with respect to outcomes of communication actions reduces the number of online replannings (as observed for sensing actions [39, 55]), and thus provides a more natural communication and collaboration with the human. (iii) Including human preferences in the planning improves human satisfaction. (iv) In connection with these physical implementations, human subject experiments with objective and subjective measures indicate that subjects find the interaction nor stressful neither frustrating. The whole collaboration was perceived to be safe with sufficient verbal communication, and the team completed the tasks in the presence of communication. (v) Statistically significant improvement in task completion times, safety have been shown when communication was present.

8.1 Contributions

In this work, we proposed a formal framework for collaborative assembly planning, under uncertainty for a human-robot team. Our contributions are as follows:

1. We introduced a formal method that allows planning of sensing actions and

various hybrid communication actions, in addition to hybrid actuation actions.

2. We formally modeled an assembly planning problem to resolve the uncertainty caused by the incomplete knowledge about the world state.

3. We formally represented the collaborative assembly planning which is an extension of our existing assembly planning formulation. As human is involved in this environment setting, uncertainty increases due to the incomplete knowledge about the human physical and mental state.

4. We plan for communication, in addition to actuation and sensing actions not only for providing evidence-based explanations to humans but also for safer collaborations. Communication actions are formalized according to their types, such as:

    - The robot asking/offering human help
    - The robot requesting human to perform an action
    - The robot communicating to confirm human intention of performing some task
    - The robot initiating/ending conversations.

5. Safety and verbosity levels are introduced as human preferences.

6. The proposed method utilizes commonsense knowledge not only for ensuring the assembly plans to be natural and to improve human acceptance.

7. The formal hybrid conditional planning framework can also provide optimized plan length.

8. We performed experimental evaluations of the proposed approach and tested the efficacy of our framework using dynamic simulations and human subject

experiments. The results indicate that collaboration with planned communication actions between human and robot are statistically significantly safer and have better success rate.

8.2 Future Work

For future directions, we plan to extend our approach to include other types of human-robot communications, like gestures, in the spirit of [24], to further improve collaborative planning.

Another topic that can be explored is the integration of learning methods to learn from the interaction experience. The robot can learn the preferences of human with different age groups. For instance, a 40 year old person may not show active collaboration, whereas a 20 year old like to be proactive. An older person may prefer a collaboration where the robot is too close to human. Such traits can be learned through experience and may be helpful for more natural collaboration.

Furthermore, we would like to extend our framework to consider several teammates. For example, a robot working at a restaurant counter, he needs to collaborate with the waiter (robot or human) to perform actuation actions to reach a common goal that is serving a customer, also the robot may need to interact with other customers through communication. Planning for such a scenario requires the robot to be social enough to deal with customers with whom there is no collaboration, but also to be collaborative with the waiter to help her complete the orders.

References

[1] Rana G Ayoub and Keith L Doty. A representation for discrete assembly sequences in task planning. In *Computer Software and Applications Conference, 1989. COMPSAC 89., Proceedings of the 13th Annual International*, pages 746–753. IEEE, 1989.

[2] Chitta Baral, Vladik Kreinovich, and Raul Trejo. Computational complexity of planning and approximate planning in presence of incompleteness. In *Proc. of IJCAI*, pages 948–955, 1999.

[3] Gerhard Brewka, Thomas Eiter, and Miroslaw Truszczynski. Answer set programming: An introduction to the special issue. *AI Magazine*, 37(3):5–6, 2016.

[4] Stephane Cambon, Rachid Alami, and Fabien Gravot. A hybrid approach to intricate motion, manipulation and task planning. *The International Journal of Robotics Research*, 28(1):104–126, 2009.

[5] Li Da Xu, Chengen Wang, Zhuming Bi, and Jiapeng Yu. Autoassem: an automated assembly planning system for complex products. *IEEE Transactions on Industrial Informatics*, 8(3):669–678, 2012.

[6] Neil T. Dantam, Zachary K. Kingston, Swarat Chaudhuri, and Lydia E. Kavraki. Incremental task and motion planning: A constraint-based approach. In *Proc. of RSS*, 2016.

[7] LS Homem De Mello and Arthur C Sanderson. A correct and complete algorithm for the generation of mechanical assembly sequences. *IEEE transactions on Robotics and Automation*, 7(2):228–240, 1991.

[8] Thomas Eiter, Giovambattista Ianni, Roman Schindlauer, and Hans Tompits. A Uniform Integration of Higher-Order Reasoning and External Evaluations

in Answer-Set Programming. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 90–96, 2005.

[9] Thomas Eiter, Giovambattista Ianni, Roman Schindlauer, and Hans Tompits. dlvhex: A system for integrating multiple semantics in an answer-set programming framework. *WLP*, 6:206–210, 2006.

[10] Esra Erdem, Kadir Haspalamutgil, Can Palaz, Volkan Patoglu, and Tansel Uras. Combining high-level causal reasoning with low-level geometric reasoning and motion planning for robotic manipulation. In *Proc. of ICRA*, pages 4575–4581, 2011.

[11] Esra Erdem, Erdi Aker, and Volkan Patoglu. Answer set programming for collaborative housekeeping robotics: representation, reasoning, and execution. *Intelligent Service Robotics*, 5:275–291, 2012.

[12] Esra Erdem, Volkan Patoglu, and Peter Schüller. A systematic analysis of levels of integration between high-level task planning and low-level feasibility checks. *AI Commununications*, 29(2):319–349, 2016.

[13] Martin Gebser, Roland Kaminski, Benjamin Kaufmann, Max Ostrowski, Torsten Schaub, and Sven Thiele. A users guide to gringo, clasp, clingo, and iclingo. 2008.

[14] Martin Gebser, Roland Kaminski, Benjamin Kaufmann, and Torsten Schaub. Clingo = ASP + control: Preliminary report. *CoRR*, abs/1405.3694, 2014. URL `http://arxiv.org/abs/1405.3694`.

[15] Michael Gelfond and Vladimir Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Computing*, 9:365–385, 1991.

[16] Manuel Giuliani, Ronald Petrick, Mary Ellen Foster, Andre Gaschler, Amy Isard, Maria Pateraki, and Markos Sigalas. Comparing task-based and socially intelligent behaviour in a robot bartender. In *Proceedings of the 15th*

ACM on International conference on multimodal interaction, pages 263–270. ACM, 2013.

[17] Matthew C Gombolay, Reymundo A Gutierrez, Shanelle G Clarke, Giancarlo F Sturla, and Julie A Shah. Decision-making authority, team efficiency and human worker satisfaction in mixed human–robot teams. *Autonomous Robots*, 39(3):293–312, 2015.

[18] Matthew C Gombolay, Cindy Huang, and Julie A Shah. Coordination of human-robot teaming with human task preferences. In *AAAI Fall Symposium Series on AI-HRI*, volume 11, page 2015, 2015.

[19] Elena Corina Grigore and Brian Scassellati. Constructing policies for supportive behaviors and communicative actions in human-robot teaming. In *Human-Robot Interaction (HRI), 2016 11th ACM/IEEE International Conference on*, pages 615–616. IEEE, 2016.

[20] Dan Halperin, J-C Latombe, and Randall H Wilson. A general framework for assembly planning: The motion space approach. *Algorithmica*, 26(3-4): 577–601, 2000.

[21] Keliang He, Morteza Lahijanian, Lydia E. Kavraki, and Moshe Y. Vardi. Towards manipulation planning with temporal logic specifications. In *Proc. of ICRA*, pages 346–352, 2015.

[22] Frederik Heger and Sanjiv Singh. Robust robotic assembly through contingencies, plan repair and re-planning. In *Proc. of ICRA*, 2010.

[23] Hua Jiang, Yuyun Zhang, Guangleng Xiong, and Ji Zhou. Assembly sequence planning for mechanical products. *Tsinghua Science and Technology*, 4(2):1436–1439, 1999.

[24] Lars Johannsmeier and Sami Haddadin. A hierarchical human-robot interaction-planning framework for task allocation in collaborative industrial

assembly processes. *IEEE Robotics and Automation Letters*, 2(1):41–48, 2017.

[25] Leslie Pack Kaelbling and Tomás Lozano-Pérez. Integrated task and motion planning in belief space. *The International Journal of Robotics Research*, 32 (9–10):1194–1227, 2013.

[26] Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artif. Intell.*, 101(1-2):99–134, 1998.

[27] Sertac Karaman and Emilio Frazzoli. Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research*, 30 (7):846–894, 2011.

[28] Stephen G Kaufman, Randall H Wilson, Rondall E Jones, Terri L Calton, and Arlo L Ames. The archimedes 2 mechanical assembly planning system. In *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, volume 4, pages 3361–3368. IEEE, 1996.

[29] Dong-Hyung Kim, Sung-Jin Lim, Duck-Hyun Lee, Ji Yeong Lee, and Chang-Soo Han. A rrt-based motion planning of dual-arm robot for (dis) assembly tasks. In *Robotics (ISR), 2013 44th International Symposium on*, pages 1–6. IEEE, 2013.

[30] Joseph Kim, Christopher J Banks, and Julie A Shah. Collaborative planning with encoding of users' high-level strategies. In *AAAI*, pages 955–962, 2017.

[31] Ross A Knepper, Todd Layton, John Romanishin, and Daniela Rus. Ikeabot: An autonomous multi-robot coordinated furniture assembly system. In *Proc. of ICRA*, pages 855–862, 2013.

[32] Przemyslaw A Lasota and Julie A Shah. Analyzing the effects of human-aware motion planning on close-proximity human–robot collaboration. *Human factors*, 57(1):21–33, 2015.

[33] Jiming Liu. Assembly planning based on a task grammar augmented with qualitative heuristic knowledge. In *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, volume 1, pages 962–969. IEEE, 1995.

[34] JH Makaliwe and AAG Requicha. Automatic planning of nanoparticle assembly tasks. In *Proc. of ISATP*, pages 288–293, 2001.

[35] Jorge Munoz-Morera, Ivan Maza, Carmelo J Fernandez-Aguera, Fernando Caballero, and Anibal Ollero. Assembly planning for the construction of structures with multiple uas equipped with robotic arms. In *Proc. of ICUAS*, pages 1049–1058, 2015.

[36] Stefanos Nikolaidis, Jodi Forlizzi, David Hsu, Julie Shah, and Siddhartha Srinivasa. Mathematical models of adaptation in human-robot collaboration. *arXiv preprint arXiv:1707.02586*, 2017.

[37] Stefanos Nikolaidis, Minae Kwon, Jodi Forlizzi, and Siddhartha Srinivasa. Planning with verbal communication for human-robot collaboration. *ACM Transactions on Human-Robot Interaction (THRI)*, 7(3):22, 2018.

[38] Korbinian Nottensteiner, Tim Bodenmueller, Michael Kassecker, Maximo A Roa, Andreas Stemmer, Theodoros Stouraitis, Daniel Seidel, and Ulrike Thomas. A complete automated chain for flexible assembly using recognition, planning and sensor-based execution. In *ISR 2016: 47st International Symposium on Robotics; Proceedings of*, pages 1–8. VDE, 2016.

[39] Ahmed Nouman, Ibrahim Faruk Yalciner, Esra Erdem, and Volkan Patoglu. Experimental evaluation of hybrid conditional planning for service robotics. In *Proc. of ISER*, pages 692–702, 2016.

[40] Mark A Peot and David E Smith. Conditional nonlinear planning. In *Artificial Intelligence Planning Systems*, pages 189–197. Elsevier, 1992.

[41] Claudia Pérez-D'Arpino and Julie A Shah. Fast motion prediction for collaborative robotics. In *IJCAI*, pages 3988–3989, 2016.

[42] Ronald PA Petrick and Mary Ellen Foster. Planning for social interaction in a robot bartender domain. In *ICAPS*, 2013.

[43] Louise Pryor and Gregg Collins. Planning for contingencies: A decision-based approach. *JAIR*, 4:287–339, 1996.

[44] Ruth Schulz, Philipp Kratzer, and Marc Toussaint. Preferred interaction styles for human-robot collaboration vary over tasks with different action types. *Frontiers in neurorobotics*, 12:36, 2018.

[45] E Sebastiani, R Lallement, R Alami, and L Iocchi. Dealing with on-line human-robot negotiations in hierarchical agent-based task planner. In *Proc. of International Conference on Automated Planning and Scheduling (ICAPS 2017)*, 2017.

[46] Julie A Shah, Joseph H Saleh, and Jeffrey A Hoffman. Analytical basis for evaluating the effect of unplanned interventions on the effectiveness of a human–robot system. *Reliability Engineering & System Safety*, 93(8):1280–1286, 2008.

[47] Ioan A. Şucan, Mark Moll, and Lydia E. Kavraki. The Open Motion Planning Library. *IEEE Robotics & Automation Magazine*, 19(4):72–82, 2012.

[48] Ioan Alexandru Sucan, Mark Moll, and Lydia E Kavraki. The open motion planning library. *IEEE Robotics & Automation Magazine*, 19(4):72–82, 2012.

[49] Stefanie Tellex, Ross A Knepper, Adrian Li, Daniela Rus, and Nicholas Roy. Asking for help using inverse semantics. In *Proc. of RSS*, 2014.

[50] Federico Thomas and Carme Torras. Inferring feasible assemblies from spatial constraints. *IEEE Transactions on Robotics and Automation*, 8(2):228–239, 1992.

[51] Ulrike Thomas and Friedrich M Wahl. A system for automatic planning, evaluation and execution of assembly sequences for industrial robots. In *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, volume 3, pages 1458–1464. IEEE, 2001.

[52] Vaibhav V Unhelkar, Ho Chit Siu, and Julie A Shah. Comparative performance of human and mobile robotic assistants in collaborative fetch-and-deliver tasks. In *Proc. of HRI*, pages 82–89, 2014.

[53] Vaibhav V Unhelkar, Przemyslaw A Lasota, Quirin Tyroller, Rares-Darius Buhai, Laurie Marceau, Barbara Deml, and Julie A Shah. Human-aware robotic assistant for collaborative assembly: Integrating human motion prediction with planning in time. *IEEE Robotics and Automation Letters*, 3(3): 2394–2401, 2018.

[54] David H. D. Warren. Generating conditional plans and programs. In *Proc. of AISB*, pages 344–354, 1976.

[55] Ibrahim Faruk Yalciner, Ahmed Nouman, Volkan Patoglu, and Esra Erdem. Hybrid conditional planning using answer set programming. *Theory and Practice of Logic Programming*, 17(5-6):1027–1047, 2017.

[56] Yahui Yang and Zezhi Ren. Research and application of assembly planning and scheduling system for automobile assembly mes. In *Proc. of ICCIS*, pages 1206–1209, 2013.

[57] Chongjie Zhang and Julie A Shah. Co-optimizating multi-agent placement with task assignment and scheduling. In *IJCAI*, pages 3308–3314, 2016.