Towards Integrating Hierarchical Goal Networks and Motion Planners to Support Planning for Human Robot Collaboration in Assembly Cells

Vikas Shivashankar, Krishnanand N. Kaipa, Dana S. Nau, and **Satyandra K. Gupta**
Maryland Robotics Center
University of Maryland, College Park, MD.

Introduction

Low-level motion planning techniques must be combined with high-level task planning formalisms to generate realistic plans that can be carried out by humans and robots. A representative example is planning for fenceless assembly cells where robots can collaborate seamlessly with humans to perform assembly tasks. Key constituent components include assembly sequence generation (Morato, Kaipa, and Gupta 2013), task decomposition between human and robot (Kaipa et al. 2014), system state monitoring (tracking human, robot, and assembly parts) (Morato et al. 2014a), instruction generation for humans (Kaipa et al. 2012), safety (Morato et al. 2014b), and error recovery (Morato et al. 2014a). In order to enable a coherent integration among these components, a high-level planner, interleaved with motion planners, is needed at several levels of the system hierarchy.

For example, given a CAD model of a product to be assembled, motion planning methods can generate improved assembly precedence constraints (Morato, Kaipa, and Gupta 2013), which can be compiled into a high-level planning problem. Humans and robots share complimentary strengths. The planning framework can incorporate this knowledge to decompose the tasks effectively. Further, an integral planner must be able to perform plan-repair in order to handle contingencies: (1) low-level deviations in the geometric state without affecting the corresponding symbolic state (e.g., human places part in a wrong posture), which can be corrected at the motion planning level, or (2) deviations in the symbolic state (e.g., human picks incorrect part; improved alternative sequence may or may not exist), which needs to be corrected at both levels of planning.

Task planning formalisms typically used to achieve this integration are *Classical Planning* (Cambon, Alami, and Gravot 2009; Erdem et al. 2011; Dornhege et al. 2009; Burbridge and Dearden 2013) and *Hierarchical Task Network (HTN) Planning* (Kaelbling and Lozano-Pérez 2011; Hadfield-Menell, Kaelbling, and Lozano-Perez 2013; Wolfe, Marthi, and Russell 2010). Whereas Classical planning is not scalable, HTNs impose stringent completeness requirements on domain models, which are difficult to guarantee in open, dynamic environments. Recently, we developed a new planning formalism called *Hierarchical Goal Networks* (HGNs) (Shivashankar et al. 2012; 2013) that combines scalability and expressivity advantages of HTNs and heuristic-search/reasoning capabilities of Classical planning into a single framework. In this work, we exploit the advantages of HGNs to tightly integrate it with motion planners. Our aims are twofold: (1) Design a general-purpose planning-and-execution framework that combines HGN planning and execution-time plan-repair algorithms with off-the-shelf motion planners, and (2) Formulate this planning framework in the context of planning for human robot collaboration in assembly cells.

The system takes as input the planning problem P (provides descriptions of the initial state, goals to be achieved, base action models at the task-planning level, control primitives at the motion planning level, and procedures to translate between symbolic and geometric state descriptions). P is input into an *Offline Planning* module, in which HGN planners and low-level motion planners interactively synthesize an executable plan structure Π that achieves the given goals when applied from the initial system state. Π is then input into an *Execution-time Reasoner* module to (1) monitor plan execution, and (2) repair Π in case the deviations from expected state render the current plan inexecutable.

Planning Formalism

Task Planning Domain. We define the task planning model M_{TP} as a five-tuple $(V_D, V_C, \mathcal{O}, \mathcal{M}, \gamma)$. V_D is the set of discrete state variables in the domain; they evaluate to either `true`/`false` or to a discrete object in the domain. V_C on the other hand represents the set of continuous state variables in the domain, which can evaluate to a real number. $\mathcal{O}$ represents the set of primitive operator models in the domain, which are model actions that are executable in a single step at the task planning level. Each $o \in \mathcal{O}$ is a four-tuple $(\mathtt{name}(o), \mathtt{pre}(o), \mathtt{eff}(o), \mathtt{cost}(o))$. $\mathcal{M}$ represents the set of HGN methods, which models domain-specific knowledge that suggests ways to decompose goals into subgoals. Each $m \in \mathcal{M}$ is a four-tuple $(\mathtt{name}(m), \mathtt{post}(m), \mathtt{pre}(m), \mathtt{subgoals}(m))$. Finally, γ represents the state transition function. A ground instance a of an operator o is applicable in a state s if s satisfies $\mathtt{pre}(a)$; the resulting state $s' = \gamma(s, a)$ reassigns the state variables according to the assignments in $\mathtt{eff}(a)$.

A *planning problem* is a triple $P = (M_{TP}, s_0, gn)$, where M_{TP} is a task planning model, s_0 is the initial state, and gn is a network of goals that need to be accomplished.

Motion Planning. Let $\chi = \mathbb{R}^d$, $d \leq |V_C|$ be the configuration space of the system. Let χ_{obs} be the obstacle region; thus $\chi_{free} = \chi \setminus \chi_{obs}$ represents the obstacle-free space. A motion planning problem is a triple $P = (\chi_{free}, x_0, \chi_{goal})$ where x_0 is an element of χ_{free} and the goal region χ_{goal} is a subset of χ_{free}. A path $\sigma : [0, 1] \to \mathbb{R}^d$ is a valid solution to P if (1) it is *continuous*, (2) it is *collision-free*, i.e. $\sigma(\tau) \in \chi_{free}$ for all $\tau \in [0, 1]$, and (3) the boundary conditions are satisfied, i.e. $\sigma(0) = x_0$ and $\sigma(1) \in \chi_{goal}$.

Connecting Task and Motion Planning. For this purpose, we must first provide a way to switch between these two state spaces by generating: (1) candidate geometric states consistent with a symbolic state, and (2) symbolic state corresponding to a given geometric state. We assume the following domain-specific procedures: (1) Gen_{sym} which takes as input a geometric state s_{geom} and generates the corresponding symbolic state s_{sym}, and Gen_{geom} which takes as input a symbolic state s_{sym} and generates a candidate geometric state s_{geom}.

Thus, the overall planning problem P is a 3-tuple $(\langle M_{TP}, \chi_{free}, \mathsf{Gen}_{sym}, \mathsf{Gen}_{geom}\rangle, s_0, gn)$ where s_0 and gn are the initial state and the goal network respectively. The definition of solutions for P is as follows. Let π_{sym} be a solution of the underlying HGN planning problem $P_{sym} = (M_{TP}, s_0, gn)$ (Shivashankar et al. 2013): **[Case 1]** If π_{sym} is empty, then $\Pi = \langle\rangle$ is a solution for P. **[Case 2]** Let $\pi_{sym} = a \circ \pi'_{sym}$. Furthermore, let s_1^{sym} be the symbolic state after a is executed. Let s_0^{geom} be the projection of s_0 onto the variables in V_C, and s_1^{geom} be the geometric state generated by Gen_{geom} for s_1. If there exists a valid solution σ to the motion planning problem $(\chi_{free}, s_0^{geom}, s_1^{geom})$ and Π' is a solution to $P' = (\langle M_{TP}, \mathsf{Gen}_{sym}, \mathsf{Gen}_{geom}\rangle, \langle s_1^{sym}, s_1^{geom}\rangle, gn)$, then $\Pi = \sigma \circ \Pi'$ is a solution to P.

Planning Algorithm

We have developed an integrated task-and-motion planning algorithm that combines GoDeL, a HGN planning algorithm (Shivashankar et al. 2013) with heuristic search motion planners (Likhachev et al. 2008; Likhachev and Stentz 2008). The algorithm takes as input a planning problem $P = (D, s_0, gn)$ and does the following: (1) it recursively decomposes the given goals using the given HGN methods until a primitive action a can be applied, (2) if a is to be executed by a robot, we further refine it into a motion plan by sampling a goal configuration c_g consistent with a's effects and running the motion planner (MP) on that.

The novelty in this approach comes from the particular way in which the planners are integrated. GoDeL, when invoking MP, also passes to it an upper bound $\tau\mathtt{cost}(a)$, where τ is a user-specified tolerance parameter. MP, being a heuristic search planner, can detect when the lower bound on the best possible solution it can generate exceeds this bound, and can return failure at that point. This is especially useful in cases when a bad goal configuration has been sampled,

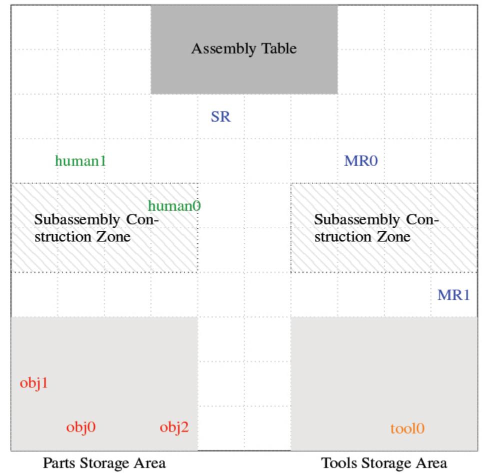

Figure 1: `MR0` and `MR1` are mobile robots, `human0` and `human1` are humans, and `SR` is a static robot. The goal is to assemble `obj0`, `obj1` and `obj2` together using `tool0`.

leading to unsolvable problems. Similarly, motion plan costs are also propagated to the task-planning level to update the heuristic estimates of the symbolic actions, and used to determine whether other actions might lead to better plan costs.

Example Domain

One of our aims is to ground the proposed planning architecture in a manufacturing domain (Fig. 1). The shop floor is divided into four regions: (1) Part storage, (2) Tool storage, (3) Subassemblies building, and (4) Final assembly.

We model the actions of the domain in **P**lanning **D**omain **D**escription **L**anguage (PDDL, a language to encode task-level planning domain descriptions) and the domain specific knowledge using HGN methods. The proposed system performs the following: **(1)** takes the product CAD model and generates assembly precedence constraints (Morato, Kaipa, and Gupta 2013) and compiles them into an HGN planning problem. **(2)** Offline planning (Use the proposed planning algorithm to generate an executable plan structure $\Pi-$ actions to be performed by humans can be provided at the symbolic level, while those performed by robots need to be refined into low-level motion plans. Protocols that need to be followed by agents (such as reserving a tool before using it, etc) can be modeled as HGN methods), and **(3)** Execution-time Reasoning (execute Π, while continuously monitoring the geometric state of the system using sensors, both vision-based and otherwise, that monitor locations of workers, tools and parts, as well as other information such as battery levels of robots, stress in robot arms while carrying heavy loads, etc. The system will repair Π using HGN and motion plan repair algorithms, when deviations from both the expected symbolic and geometric states are observed.

We believe that the techniques presented in the paper can address the above mentioned planning problems and we are currently in the process of prototyping the system.

Acknowledgements

This work was supported in part by ARO grant W911NF1210471 and ONR grants N000141210430 and N000141310597.

References

Burbridge, C., and Dearden, R. 2013. An approach to efficient planning for robotic manipulation tasks. In *ICAPS*.

Cambon, S.; Alami, R.; and Gravot, F. 2009. A hybrid approach to intricate motion, manipulation and task planning. *I. J. Robotic Res.* 28(1):104–126.

Dornhege, C.; Eyerich, P.; Keller, T.; Trüg, S.; Brenner, M.; and Nebel, B. 2009. Semantic attachments for domain-independent planning systems. In *ICAPS*.

Erdem, E.; Haspalamutgil, K.; Palaz, C.; Patoglu, V.; and Uras, T. 2011. Combining high-level causal reasoning with low-level geometric reasoning and motion planning for robotic manipulation. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 4575–4581.

Hadfield-Menell, D.; Kaelbling, L. P.; and Lozano-Perez, T. 2013. Optimization in the now: Dynamic peephole optimization for hierarchical planning. In *ICRA*.

Kaelbling, L. P., and Lozano-Pérez, T. 2011. Hierarchical task and motion planning in the now. In *ICRA*, 1470–1477.

Kaipa, K. N.; Morato, C.; Zhao, B.; and Gupta., S. K. 2012. Instruction generation for assembly operations performed by humans. In *ASME Computers and Information Engineering Conference*.

Kaipa, K. N.; Morato, C.; Liu, J.; and Gupta., S. K. 2014. Human-robot collaboration for bin-picking tasks to support low-volume assemblies. In *Human-Robot Collaboration for Industrial Manufacturing Workshop, held at Robotics: Science and Systems Conference (RSS 2014)*.

Likhachev, M., and Stentz, A. 2008. R* search. In *AAAI*, 344–350.

Likhachev, M.; Ferguson, D.; Gordon, G. J.; Stentz, A.; and Thrun, S. 2008. Anytime search in dynamic graphs. *Artif. Intell.* 172(14):1613–1643.

Morato, C.; Kaipa, K. N.; Liu, J.; and Gupta., S. K. 2014a. A framework for hybrid cells that support safe and efficient human-robot collaboration in assembly operations. In *ASME Computers and Information Engineering Conference*.

Morato, C.; Kaipa, K. N.; Zhao, B.; and Gupta, S. K. 2014b. Toward safe human robot collaboration by using multiple kinects based real-time human tracking. *ASME Journal of Computing and Information Science in Engineering* 14(1):011006.

Morato, C.; Kaipa, K. N.; and Gupta, S. K. 2013. Improving assembly precedence constraint generation by utilizing motion planning and part interaction clusters. *Computer-Aided Design* 45(11):1349 – 1364.

Shivashankar, V.; Kuter, U.; Nau, D. S.; and Alford, R. 2012. A hierarchical goal-based formalism and algorithm for single-agent planning. In *AAMAS*, 981–988.

Shivashankar, V.; Alford, R.; Kuter, U.; and Nau, D. S. 2013. The godel planning system: A more perfect union of domain-independent planning and hierarchical planning. In *IJCAI*.

Wolfe, J.; Marthi, B.; and Russell, S. J. 2010. Combined task and motion planning for mobile manipulation. In *ICAPS*, 254–258.