A MIP-Based Approach for Multi-Robot Geometric Task-and-Motion Planning

Goal : Move Objects M to Region Re

Initial setup : n_R robots, fixed objects F, n_M movable objects, n_{Re} regions. , $Gr = \bigcup Gr_{M,R}$ (object, robot)

Grounded joint action : at time j, $s_j = \langle (a^j_{R_1}, \xi^j_{R_1}) \ldots (a^j_{R_n}, \xi^j_{R_n}) \rangle$ — a : action (pick+place / wait), ξ : trajectory.

Pick & Place action : $a = \langle M, Re, R^{pick}, R^{place}, g^{pick}, g^{place}, P_M^{place} \rangle$ (object, region, robot, robot, grasp, grasp, pose)

Partially grounded joint action : $\langle \bar{a}_{R_1}, \ldots, \bar{a}_{R_n} \rangle$, $\bar{a}_{R_i} \setminus P_M^{place}$

Task skeleton $\bar{S}$: a sequence of partially grounded joint action $\xrightarrow{goal}$ S (grounded)

Swept volume : $V_{pick}(M, g, R, \xi)$ (object, grasp, robot, trajectory) , $V_{place}(M, g, R, P_M^{place}, \xi)$ (pose)

Two-Phase Method :

Phase I : Computing Collaborative Manipulation Information.

Goal : Determine all occlusion (collision-free action) and reachability information.

5 Predicates:

1. OccludePick(M_1, M_2, g, R) iff $M_1 \in V_{pick}$
2. OccludeGoalPlace(M_1, M_2, Re, g, R) iff $M_1 \in V_{place}$
3. ReachablePick (M, g, R) R can reach M
4. ReachablePlace (M, Re, g, R) R can reach Re
5. EnableGoalHandover (M, g_1, g_2, R_1, R_2) iff $R_1 \rightarrow \cdot \leftarrow R_2$ and $M \in Goal$.

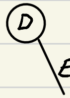
Method : 1. Use inverse-kinematics solvers and motion planner to generate ξ.

2. Minimize collision for ξ (minimum constraint removal). (costly)

3. Instead, First find ξ for movable + fixed. If failed, only ξ for fixed.

Phase II. Searching for Task-and-Motion Plans

Goal: find high quality task-and-motion plans.

Search Tree: 

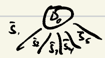
D → sequence of grounded joint actions

E → task skeleton
Value: cumulated reward

Case 1: Grounding is successful

Case 2: not successful → generate new task skeleton for objects (key Component I)

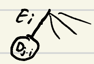
Monte-Carlo Tree Search:

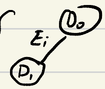
1. Initialize a set of task skeletons for $M \in G$ 

   Add root node D_0

2. Selection: select E with highest $UCB = \left(\frac{E_i.\text{value}}{E_i.\text{visits}+1} + c \times E_i.\text{prior} \times \frac{\sqrt{D_j.\text{visits}}}{E_i.\text{visits}+1} \right)$

   (value: cumulated reward; visits: # of visits; c: constant; prior: bias)

3. Expansion: create a new node $D_{j.i}$ → head node of E_i 

4. Evaluation: Ground 

   i. failed, set $r = 0$.

   ii. founded, $r = 1 + \alpha \frac{1}{|\bar{s}^*|}$ ($\alpha = 1$) → # of objects to move

   iii. founded $S' + M^*$, need to generate skeleton to move M^*

   If no plan found, $r = 0$

   else $r = \frac{S'.\text{length}}{S'.\text{length} + \bar{s}^*.\text{length}} + \alpha \frac{1}{|S'| + |\bar{s}^*|}$

   (length: # of time steps; $\bar{s}^*$: best s to move M^*)

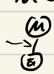
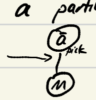
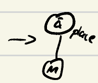
5. Backpropagation

   Update $E^{\text{selected}}.\text{value} \mathrel{+}= r$, num_visits ++.

Key Component 1: Generating Promising Task Skeletons.

- Building the collaborative manipulation task graph

  Two types of nodes: object node $M \in \mathcal{M}$, action node $\bar{a}$ partially grounded.

  Three types of edges: action edge → , block-pick edge →  block-place edge → 

• AddObject Algorithm

For each R and grasp, construct $\bar{a}$ and check for reachability

If M is in G, then compute potential handover actions:

for each $\bar{a}$, add $\bar{a}$ to C.actionnodes, and $M \to \bar{a}$ to C.actionedges.

for every M

If OccludesPick( ), AddObject(M, C) and C.block_pick_edges.add $\bar{a} \to M$

If $M \in G$,

for every M,

if occupiesGoalPlace, AddObject(M, C), C.block_place.edges.add $\bar{a} \to M$

Result: construct a full CMTG for later tasks.

• MIP formulation and Solving

Goal: find the best plan (minimum objects to be moved) from CMTG

Formulation: minimize $\sum X'_{M,\bar{a}}$

subject to ......

• Key Component 2: Task-Skeleton Grounding

• Reverse search (not-moved) (moved in fut)

1. Start at time T, sample collision free placements with respect to $M_{out} \cup M_{fut} \cup F \cup V_{fut}$

2. Plan pick & place trajectories collision free w.r.t $F \cup M_{fut} \cup M_{out}$

If succeed → expand V_{fut}, M_{fut}, and S_{fut}

3. Repeat for T-1 ....

4. If failed after all → relax constraints ~~M_{out}~~ ok to collide → new $\bar{s}$

5. Return S' and M^* ($M \in G$ not yet moved $\cup$ M occlude S')

6. If $|M^*| \neq 0$, generate new $\bar{s}$, else found

7. If failed afterall, return failure.