Lazy Rearrangement Planning in Confined Spaces

Rui Wang, Kai Gao, Jingjin Yu, Kostas Bekris

Computer Science Department, Rutgers University, New Jersey

Problem Domain

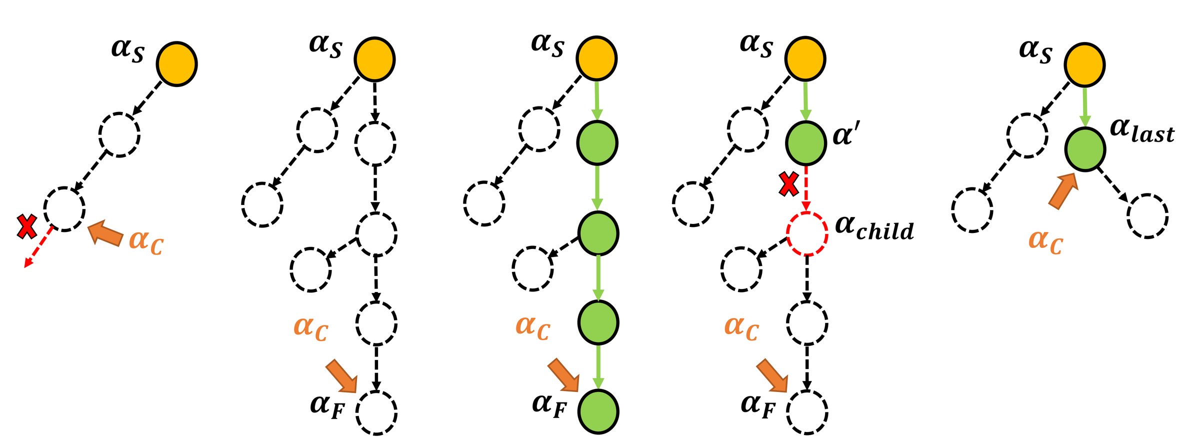
- Rearrange objects in **confined spaces**, e.g., grocery and warehouses shelves
- **Harder than tabletop setups**: Confined workspace, limited arm's reachability, and frequent undesirable collisions

General Approaches and Challenges

- Rearrangement can be modeled as a tree search problem where the goal is to search for an ordering with which objects are rearranged.
- Computational overhead lies in the pick-and-place motion planning calls, which involves collision checking.
- State-of-art methods suffers the combinatorial number of motion planning calls, which significantly limit their scalability and efficiency.

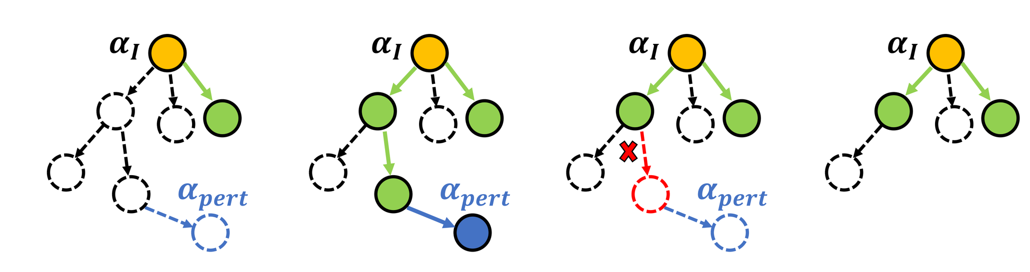
How can we find a solution quickly by avoiding unnecessary and expensive motion planning calls?

Algorithmic Insights



(a) The trees grows lazily by forward checking with constraints. (b) The tree connects start and goal arrangement lazily. (c) The branch passes path verification. (d) Path verification fails at certain edge. (e) The invalid subtree is trimmed and the search continues.

- **Contribution 1:** An **efficient, lazy local monotone solver** that
  - generates constraints from grasp reachability
  - builds a lazy search tree that respects these constraints
  - only performs path verification when a promising solution is found
- **Contribution 2:** A **global, non-monotone global planner** that
  - loads lazily the local tree lazily into the global tree framework
  - checks edges only if needed for moving an object to an intermediate position

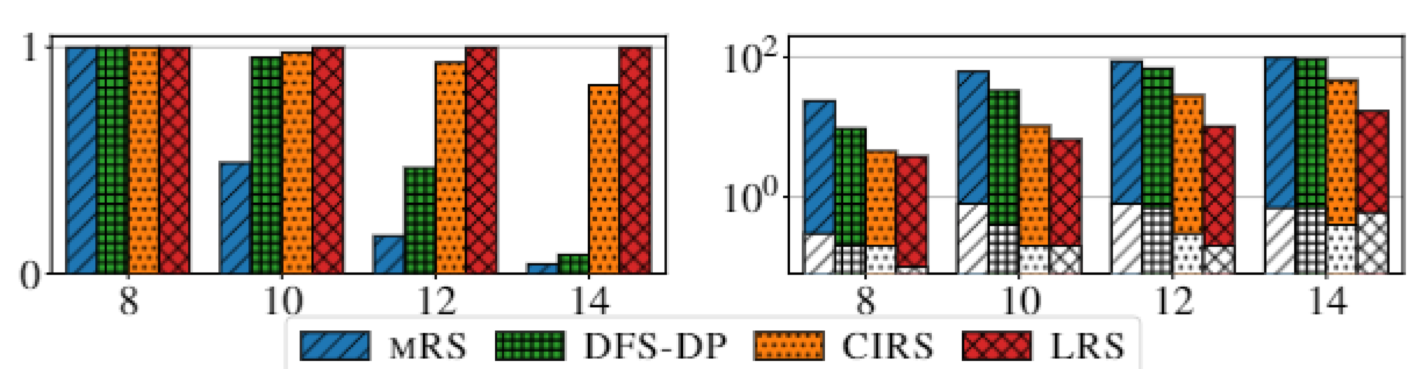


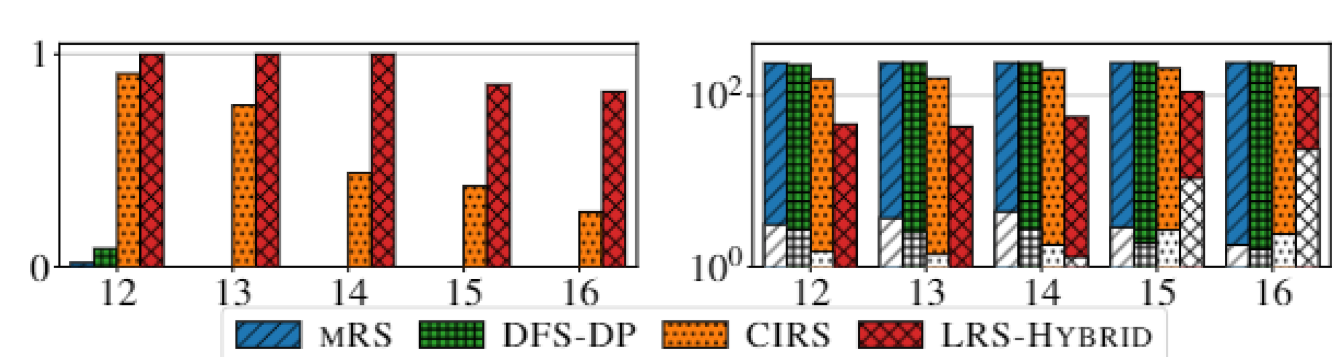
(a) A node is selected to perform perturbation on a tree. (b) A successful perturbation. (c) An unsuccessful perturbation given failure path verification. (d) The invalid branch is trimmed.

Experimental Results

The proposed method (LRS) outperforms three state-of-art methods (1) mRS, (2) DFS_{DP}, and (3) $CIRS$ in terms of

- Feasibility: **3 times higher** success rate
- Efficiency: **61% faster** computation time
- Scalability: scale up to **16 objects**





Links(paper, codes)

Acknowledgement

Supported in part by an NSF HDR TRIPODS award 1934924