Multi-object Rearrangement

Daniel Nakhimovich, Jingjin Yu and Kostas E. Bekris
Rutgers, the State University of New Jersey

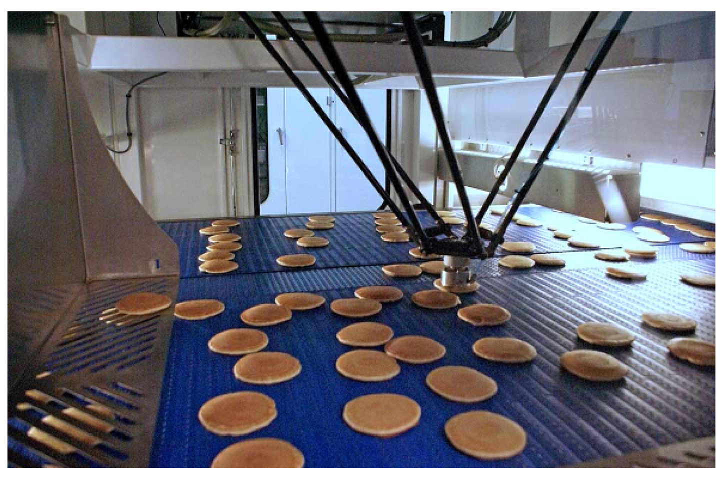
Motivation

Real Problems

- Overhand rearrangement: a manipulator can reach any object and can't collide during transfer or transit.
- Multi-robot rearrangement: reachability is trivial but collisions between robots along their motion paths needs to be considered.
- Cluttered environment: a manipulator may not reach all objects immediately so some may need to be moved out of the way first.

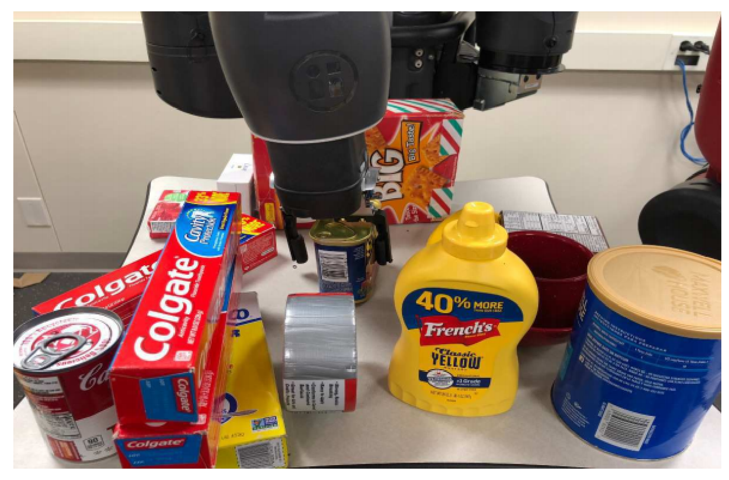
Combinatorial Challenges

- Even without overlapping congifurations, the manipulor-based multi-object rearrangement problem is NP-hard.
- The multi-robot rearrangement problem is also NP-hard even when possible paths are restricted to a discrete graph.
- For cluttered rearrangement, the end-effector pose has to be sampled from $SE(3)$ for each start and goal object placement.



https://www.youtube.com/watch?v=wg8YYuLLoM0

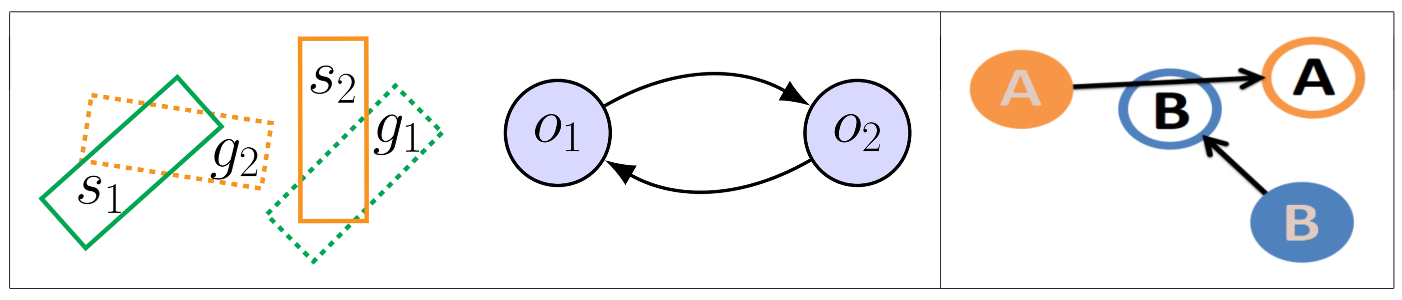
https://www.dailymail.co.uk/sciencetech/article-2855570/Amazons-new-robot-army-ready-ship.html



Problem Representation

Dependency

- Object A is dependent on B if B needs to be moved before A.
- Dependencies can either be path agnostic or path sensitive.
- In general there are $(2^{2(n-1)})^n$ possible dependency graphs for a problem instance with n objects.



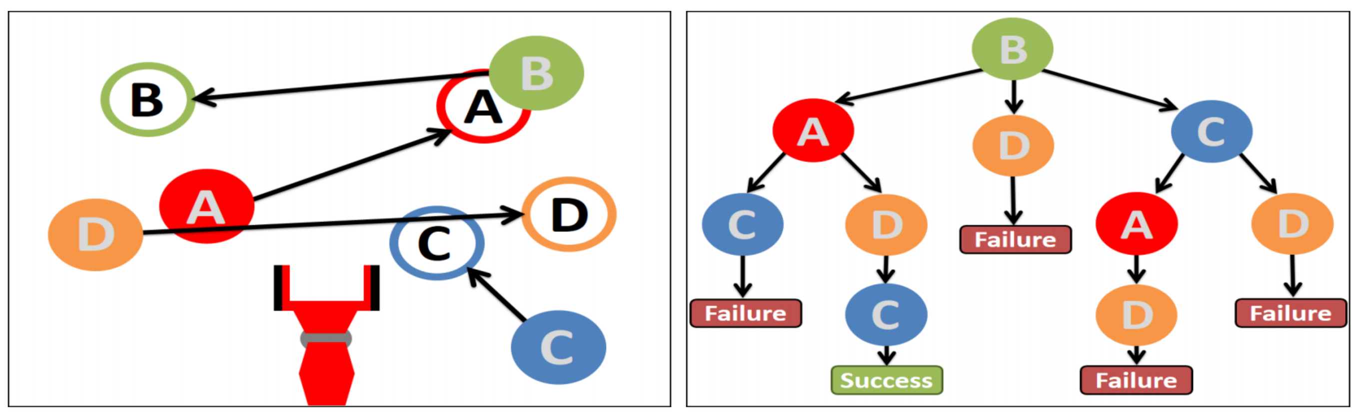
o_1/o_2 have path agnostic dependencies to each other due to overlapping start/goal states.
A is path dependent on B because the straight path for A collides with the goal of B.

Algorithmic Approaches

Integer Linear Programming: Many instances of rearangement problems can be reformulated and solved by exisiting ILP solvers.

Fast Monotone Strategy:

1. Find a path between all start and goal configurations.
2. Build a graph from path agnostic and sensitive dependencies.
3. Topologically sort the nodes of the dependency graph.

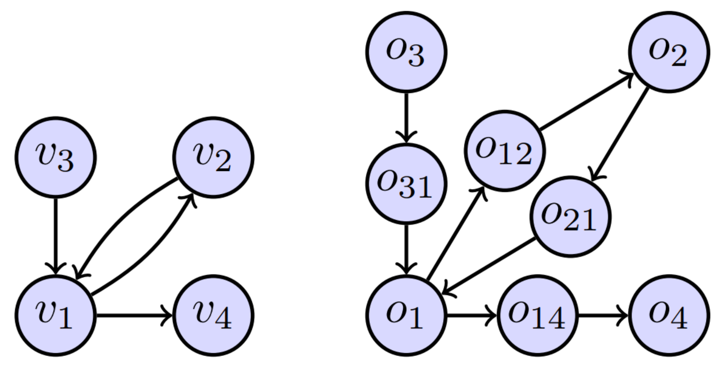


Non-monotone Extension: Solutions between intermediate workspace configurations can be composed via a tree search.

Algorithmic Insights

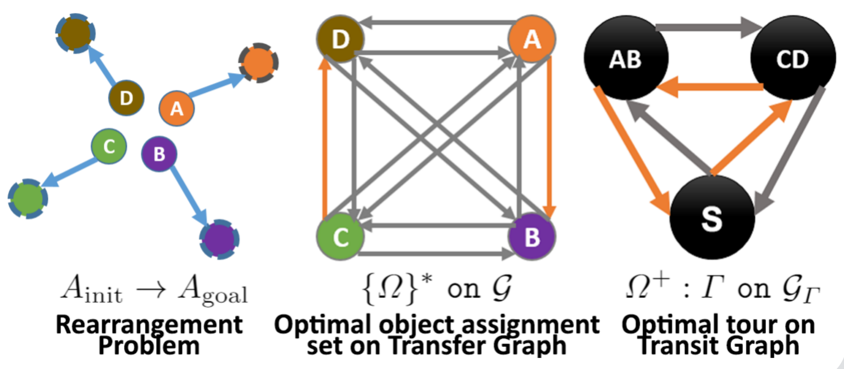
Dependency Resolution

- Feedback Node Set: node set whose removal makes graph acyclic.
- Finding minimum FNS has one-to-one correspondance to resolving dependencies cycles minimally.



Dual-manipulator Rearrangement

- Using two arms in parallel often produces faster object rearrangement solutions.
- Pairwise object matchings yield optimal motion sequences.



Research Questions

- How can we better identify optimal dependency graphs while avoiding expensive path computations?
- Can we use simultaneous manipulators to solve non-monotone problems?
- How can we use data-driven approaches to inform dependency resolution?

Acknowledgements

Supported through the NSF CISE award CCF-1934924

References

[Han+18] S. Han et al. "Complexity Results and Fast Methods for Optimal Tabletop Rearrangement with Overhand Grasps". In: *International Journal of Robotics Research* (2018).

[KB16] A. Krontiris and K. Bekris. "Efficiently Solving General Rearrangement Tasks: A Fast Extension Primitive for an Incremental Sampling-based Planner". In: *International Conference on Robotics and Automation (ICRA)*. 2016.

[KSB19] A. Kimmel, R. Shome, and K. Bekris. "Anytime Motion Planning for Prehensile Manipulation in Dense Clutter". In: *Advanced Robotics* (2019).

[Sho+18] R. Shome et al. "Fast, High-Quality Dual-Arm Rearrangement in Synchronous, Monotone Tabletop Setups". In: *International Workshop on the Algorithmic Foundations of Robotics*. 2018.