Assembly Planning for Robots

Team 5
Kim Jaemin
Jung Geumyoung

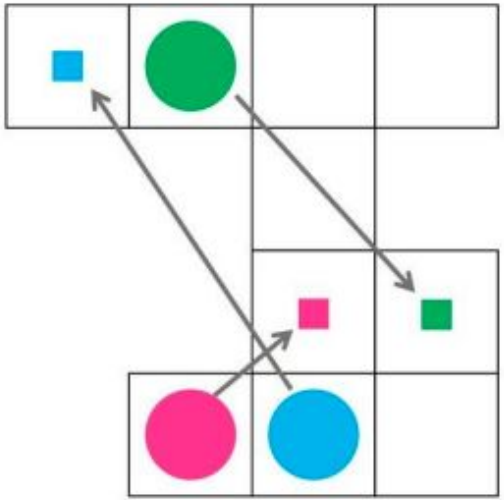
KAIST

Recap : Multi-agent path finding

Finding collision free paths for each multi-agents in the sharing environment

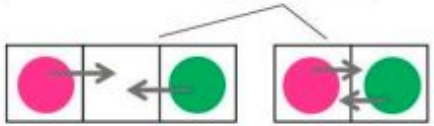
given

- graph
- agents (starts)
- goals



obtain

1. paths without collisions



s.t.

all agents are on their goals **simultaneously**

computationally **DIFFICULT** to obtain optimal solutions

Key considerations :

- conflict
- objective functions
- extensions

Approaches :

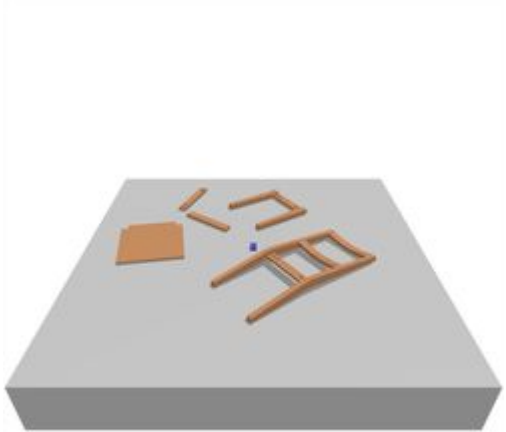
- learning-based
- search-based
- optimization-based

Contents

- Problem definition & Motivation
- Challenges
- Case study: ASAP
- Summary & Quiz

Introduction

Robotic Assembly : Robot task to manipulate and join constituent parts into a predetermined final product or sub-assembly

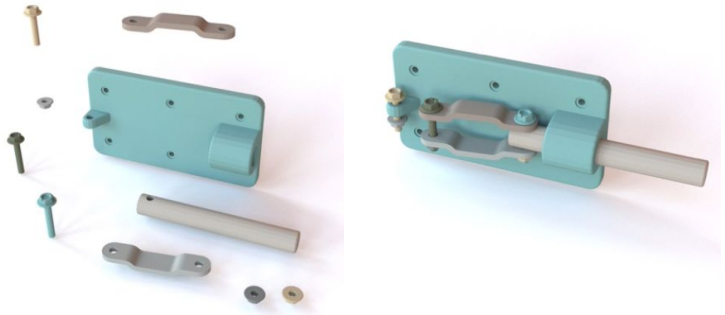


Robot assembles IKEA chair frame

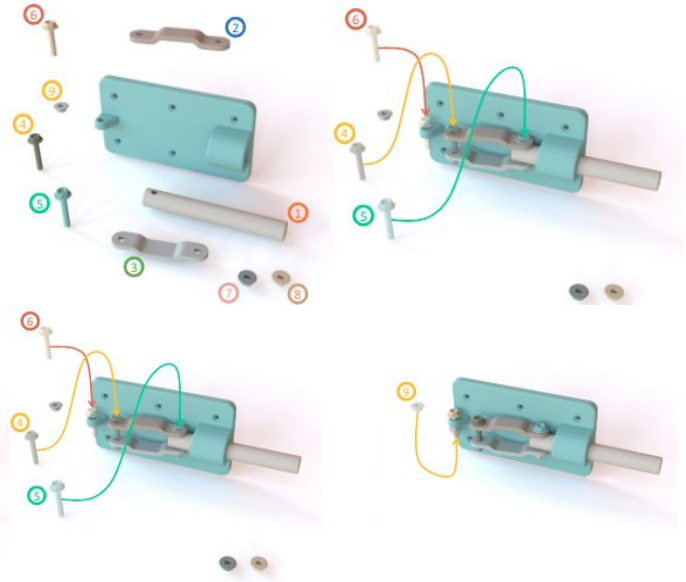
[1] Suárez-Ruiz et al., Can robots assemble an IKEA chair? *Science Robotics*, 2018
[2] Lee et al., IKEA Furniture Assembly Environment. ICRA, 2021

Problem Definition

Given: A set of 3D parts and final assembled state

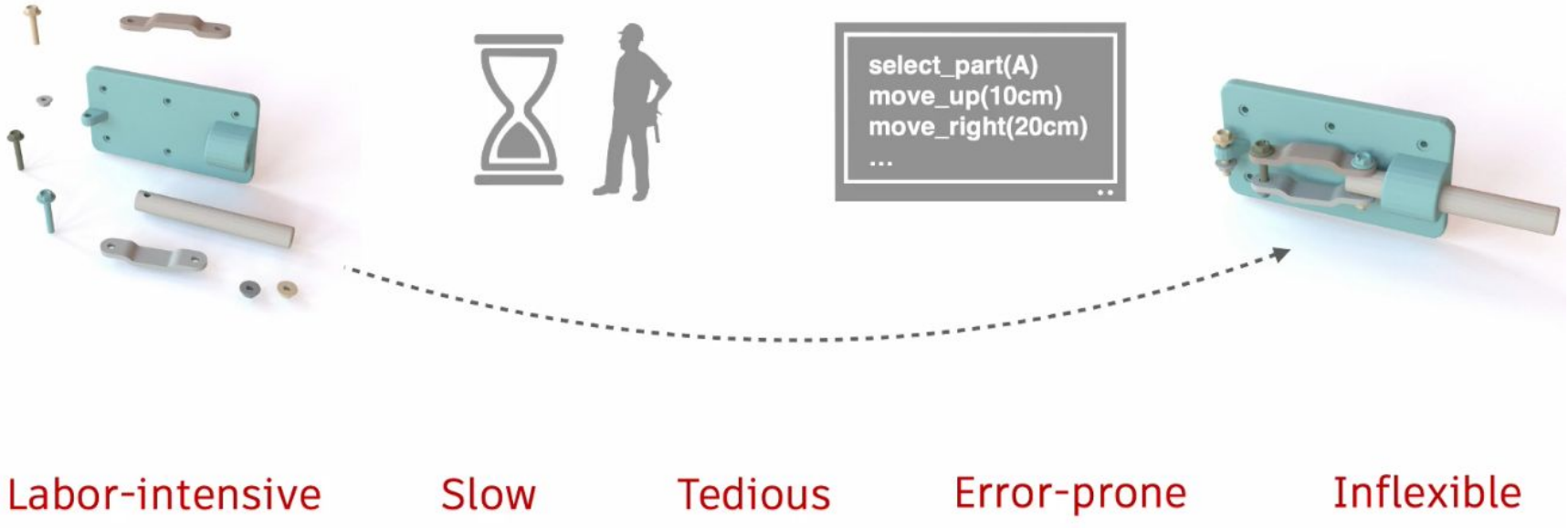


Goal: A real-world feasible, sequentially valid assembly plan



Motivation

In manufacturing industry, the assembly process is usually planned by humans with hardcoded instructions.



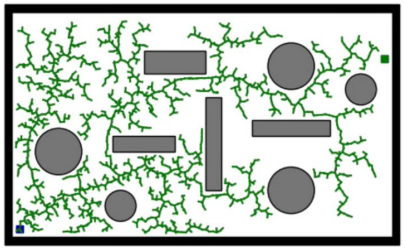
Labor-intensive Slow Tedious Error-prone Inflexible

Challenges

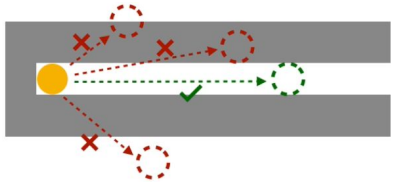
1. Part motion planning

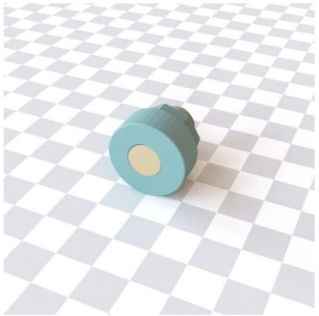
Geometric path-planning doesn't generalize well on complex assembly, because

- Large search space (6 DoF 3D poses)
- Highly constrained



RRT [LaValle 1998]



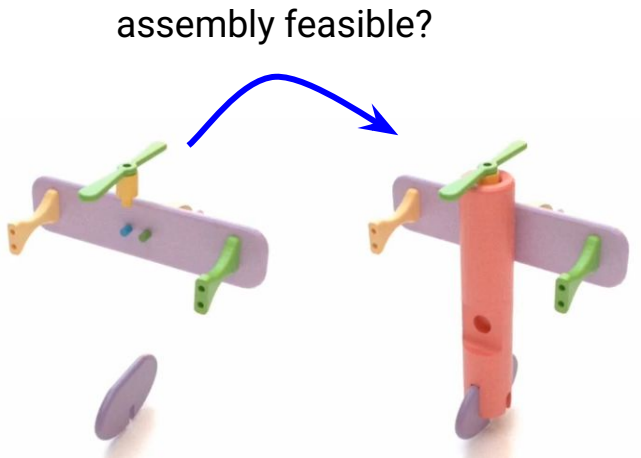


- Arbitrary shape
- "Narrow passage" problem
- Hard to deal with rotational motion

Challenges

2. Physical feasibility

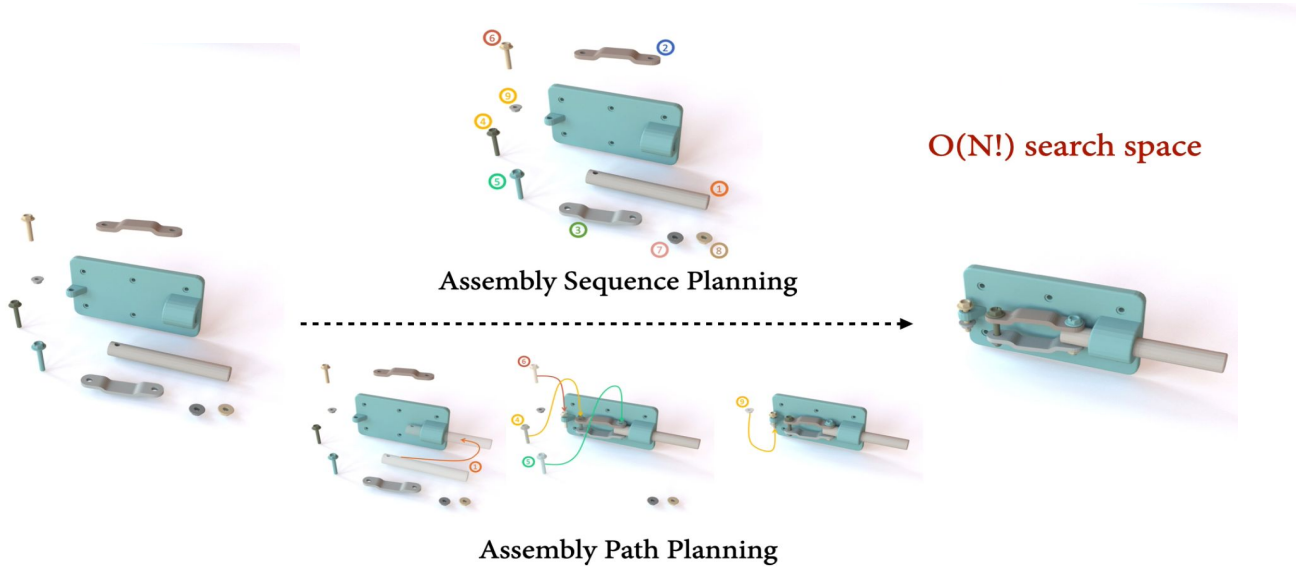
- Assembly is conducted under physical constraints (gravity)
- Must consider physical stability of parts for robotic execution



Challenges

3. Sequence planning under constraints

- Number of possible assembly sequences **grow factorially**
- Due to **precedence constraints**, only few of them are actually valid



Presents : ASAP

Automated Sequence Planning for Complex Robotic Assembly with Physical Feasibility

Input & output

Object Meshes

Assembled Poses

Robot Specifications (Optional)

ASAP

- Assembly Sequences
- Assembly Paths
- Intermediate Poses
- Parts to be Held
- Grasps & Robot Motion (optional)

Complete Assembly Plans

KAIST

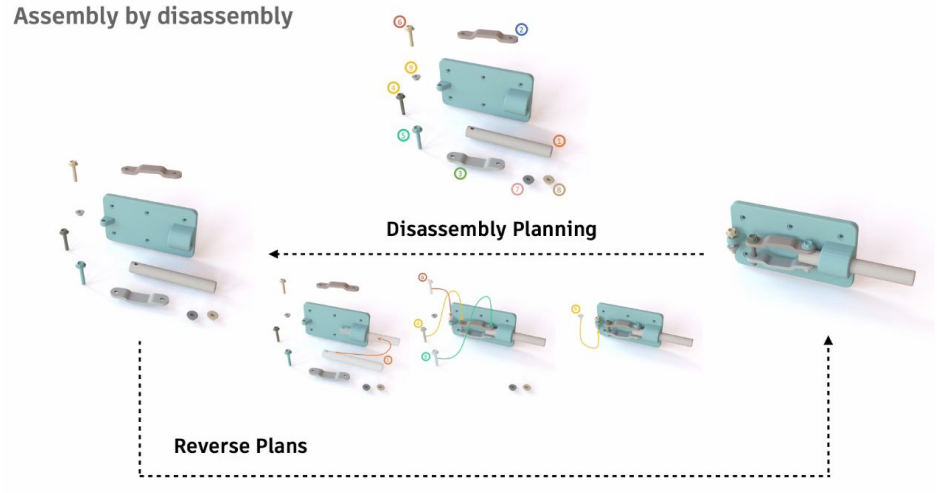
Tian et al., ASAP: Automated Sequence Planning for Complex Robotic Assembly with Physical Feasibility. ICRA, 2024

ASAP | Assembly by Disassembly

Assembly of n parts → **O(n!)** search space

Assume **rigid bodies**, then

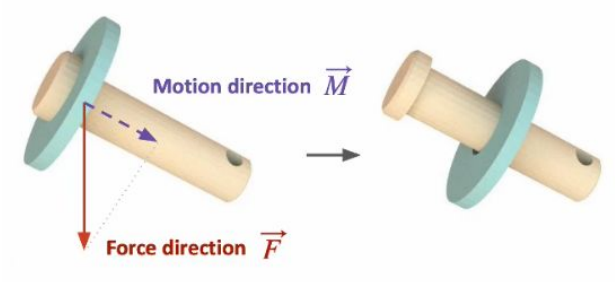
- Assembly = reversed disassembly
- Disassembly of n parts → **O(n^2)** search space (efficient!)



KAIST

ASAP | Key Components

How can I take part P out of G?

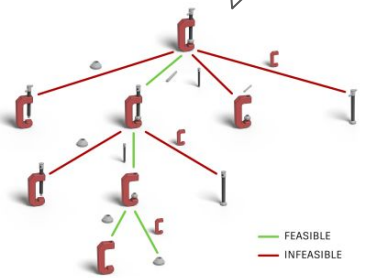


Part motion planning

Is G stable under gravity?

Stability check

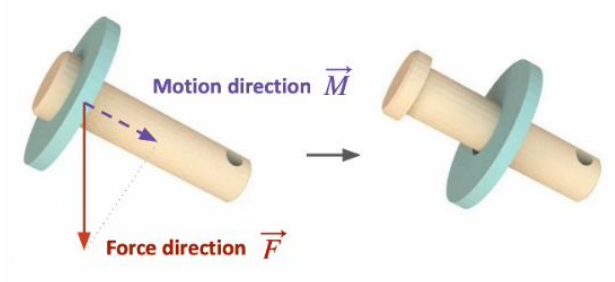
What is a viable sequence to disassemble G?



Disassembly tree search

ASAP | Key Components

How can I take part P out of G?

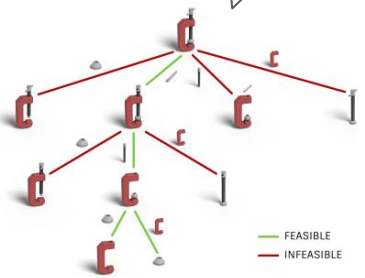


Part motion planning

Is G stable under gravity?

Stability check

What is a viable sequence to disassemble G?



Disassembly tree search

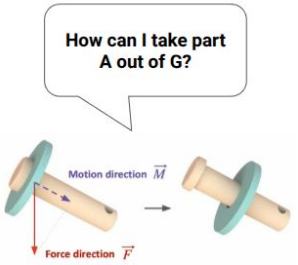
Part Motion Planning

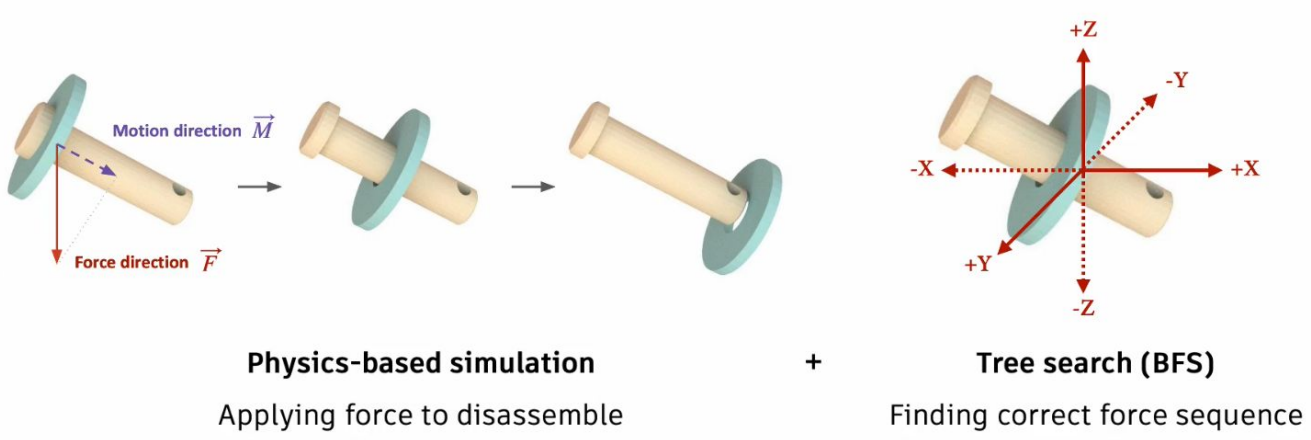
Q. “How can I take part *P* out of assembly *G*”?

Apply actions (forces / torques) in physics-based simulation

Check for **updated part poses**

Use a **tree search (BFS)** to find the action sequence





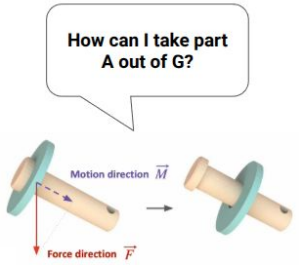
KAIST

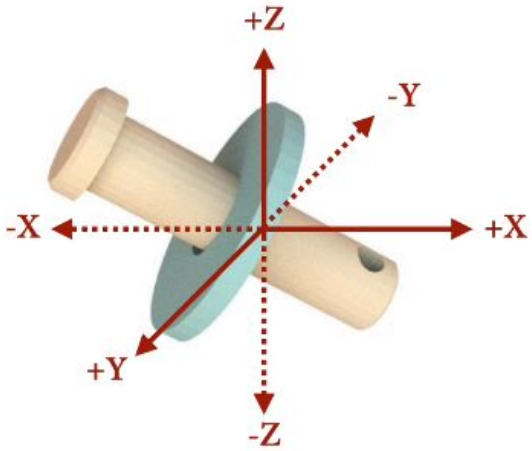
Tian et al., ASAP: Automated Sequence Planning for Complex Robotic Assembly with Physical Feasibility. ICRA, 2024

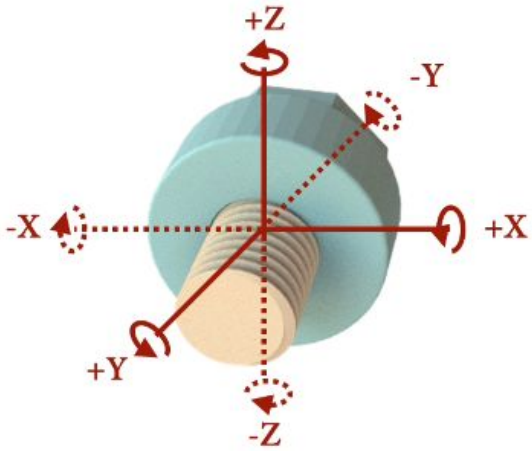
Part Motion Planning

Q. “How can I take part *P* out of assembly *G*”?

12 discrete actions (axis-aligned F & τ)





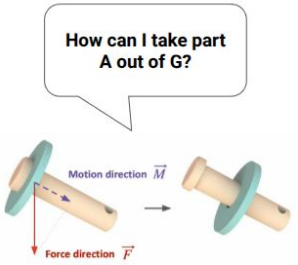


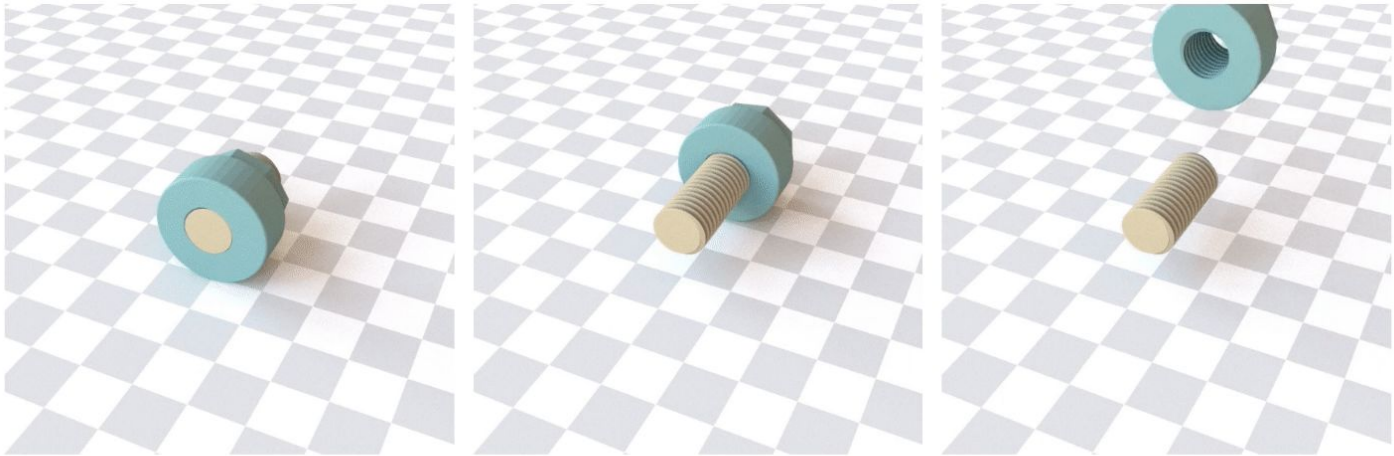
Tian et al., ASAP: Automated Sequence Planning for Complex Robotic Assembly with Physical Feasibility. ICRA, 2024

Part Motion Planning

Q. “How can I take part *P* out of assembly *G*”?

12 discrete actions (axis-aligned F & τ)

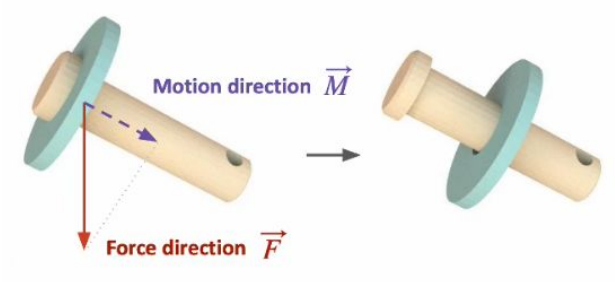




Tian et al., ASAP: Automated Sequence Planning for Complex Robotic Assembly with Physical Feasibility. ICRA, 2024

ASAP | Key Components

How can I take part P out of G?

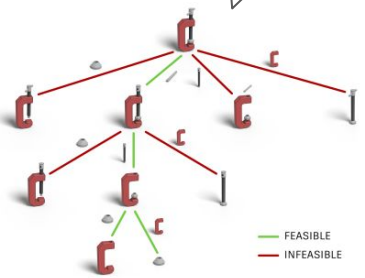


Part motion planning

Is G stable under gravity?

Stability check

What is a viable sequence to disassemble G?



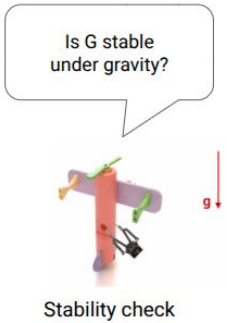
Disassembly tree search

Stability Check

Q. “Is *G* at pose *s* stable under gravity?”

Two-part problem:

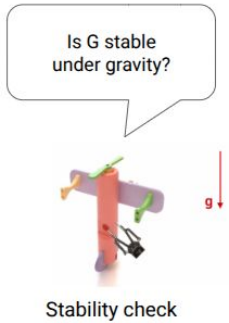
1. Finding **candidate poses**
2. Performing the **stability check** for a given pose



Stability check

Stability Check

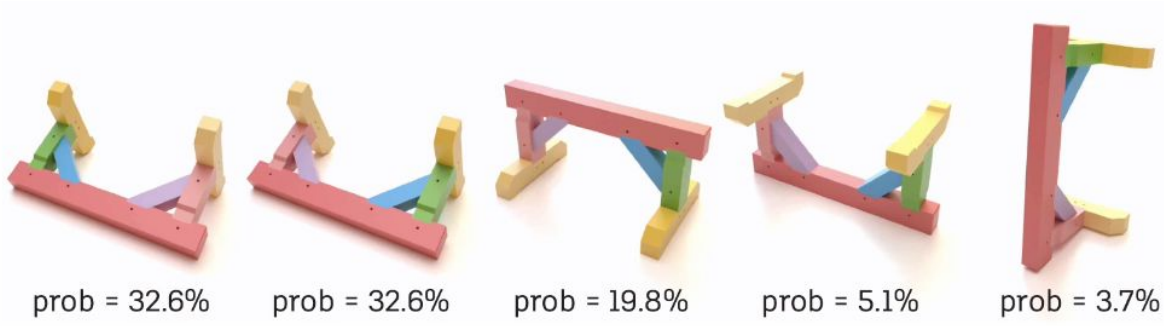
Q. “Is *G* at pose *s* stable under gravity?”



1. **Finding candidate poses**

Quasistatic pose estimator gives a few ‘good’ candidate poses

- Assumes slow, gentle placements
- Poses determined by geometry & mass distribution alone



Stability Check

Q. “Is *G* at pose *s* stable under gravity?”

2. **Stability check** for a given pose

Physics-based simulation

Check if any parts fall after certain time steps

Evaluate stability conditioned on the pose and parts to hold

Part-holding strategy

Identify which parts are to be held (by grippers/fixtures)

How to hold *N* parts by *M* fixtures?

Algorithm 2: Physics-Based Stability Check

Input: Assembly $G = \{p_1, p_2, ..., p_n\}$ with pose s, max M parts to be held, max simulation steps N, stable moving distance threshold d_{th}.

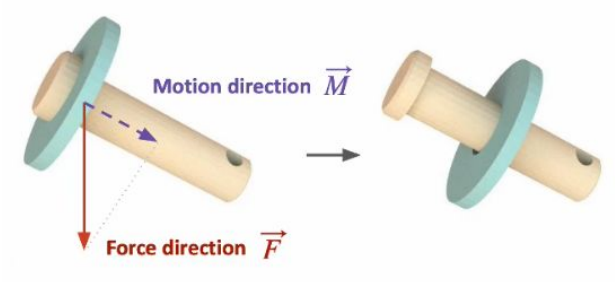
Output: Parts to be held to make G stable with pose s.

```
P_g ← {};
while |P_g| ≤ M do
    ResetSimulation(G, s, P_g);
    for i in 1, ..., n do q_{i_0} ← GetPosition(p_i);
    stable ← true;
    for j in 1, ..., N do   // N-step stability check
        Simulate(Δt);       // run physics simulation
        for i in 1, ..., n do
            q_{i_j} ← GetPosition(p_i);
            if ||q_{i_j} − q_{i_0}|| > d_th or IsDisconnected(p_i, G)
              then
                stable ← false;          // p_i falls
                P_g ← P_g ∪ {p_i};       // hold p_i
                break;
        if not stable then break;
    if stable then return P_g; // stable for N steps
return null;                   // unstable
```

KAIST

ASAP | Key Components

How can I take part P out of G?

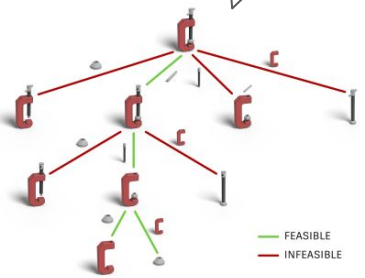


Part motion planning

Is G stable under gravity?

Stability check

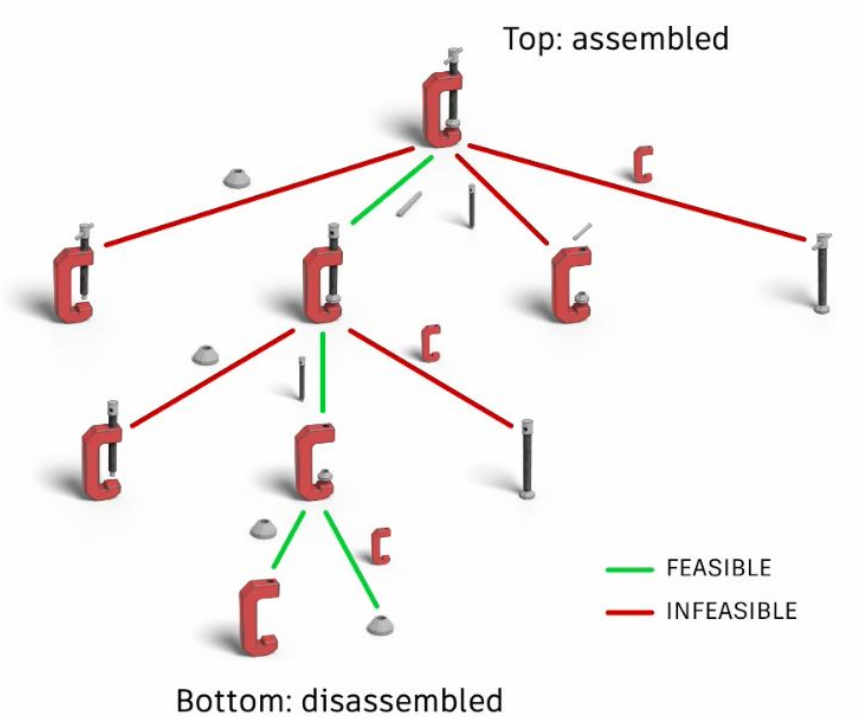
What is a viable sequence to disassemble G?



Disassembly tree search

Disassembly Tree Search

Q. “What is a viable sequence to disassemble G?”



Tian et al., ASAP: Automated Sequence Planning for Complex Robotic Assembly with Physical Feasibility. ICRA, 2024

Disassembly Tree Search

Q. “What is a viable sequence to disassemble G?”

```
def asap():
    tree.add_node(root_node)
    for node in select_node(tree):
        for part in select_part(node):
            for pose in select_pose(node):
                check_assemblable(node, part, pose)
                check_stable(node, part, pose)
                if success:
                    child_node = node \ {part}
                    tree.add_edge(node, child_node)
    return tree
```

Node selection

Part selection

Pose selection

Quasistatic Pose Estimator

Tian et al., ASAP: Automated Sequence Planning for Complex Robotic Assembly with Physical Feasibility. ICRA, 2024

Disassembly Tree Search

Q. “What is a viable sequence to disassemble G?”

```
def asap():
    tree.add_node(root_node)
    for node in select_node(tree):
        for part in select_part(node):
            for pose in select_pose(node):
                check_assemblable(node, part, pose)
                check_stable(node, part, pose)
                if success:
                    child_node = node \ {part}
                    tree.add_edge(node, child_node)
    return tree
```

Node selection

Part selection

Pose selection

Feasibility evaluation

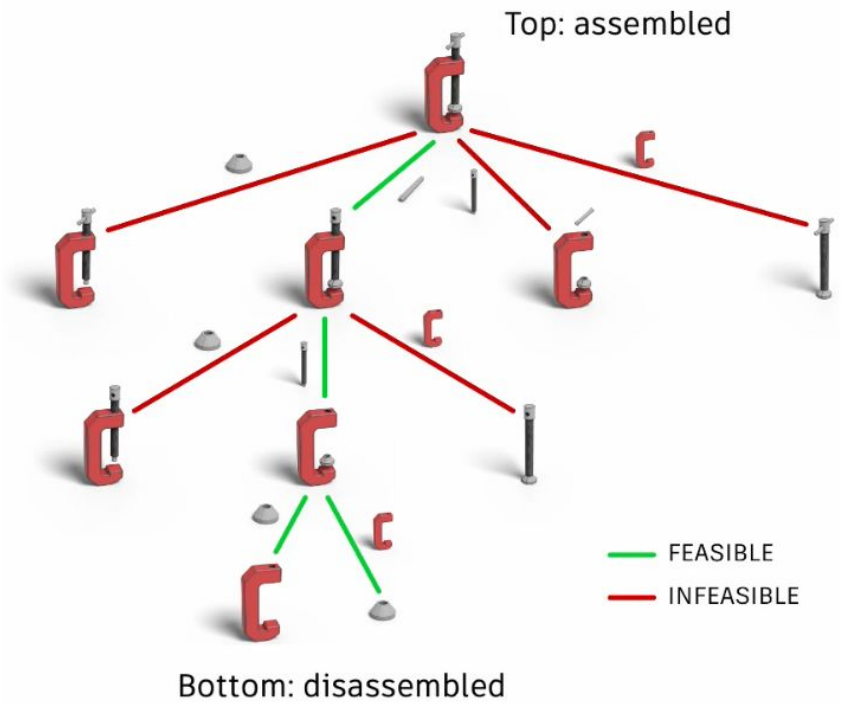
Part motion planning

Stability check

Tian et al., ASAP: Automated Sequence Planning for Complex Robotic Assembly with Physical Feasibility. ICRA, 2024

Disassembly Tree Search

Q. “What is a viable sequence to disassemble G?”




```
def asap():
    tree.add_node(root_node)
    for node in select_node(tree):                    Node selection
        for part in select_part(node):                Part selection
            for pose in select_pose(node):            Pose selection
                check_assemblable(node, part, pose)
                check_stable(node, part, pose)
                if success:                           Feasibility evaluation
                    child_node = node \ {part}
                    tree.add_edge(node, child_node)
    return tree
```


Tian et al., ASAP: Automated Sequence Planning for Complex Robotic Assembly with Physical Feasibility. ICRA, 2024

Summary

Walkthrough on robotic assembly planning pipeline

1. Part motion planning : Discrete action sampling
2. Physical stability check : Physics-based simulation
3. Assembly sequence planning : Disassembly tree search

Thank you

Quiz

1. In ASAP, which of the following is used for assembly sequence planning?

   a. RRT

   b. Disassembly tree search

   c. A* algorithm

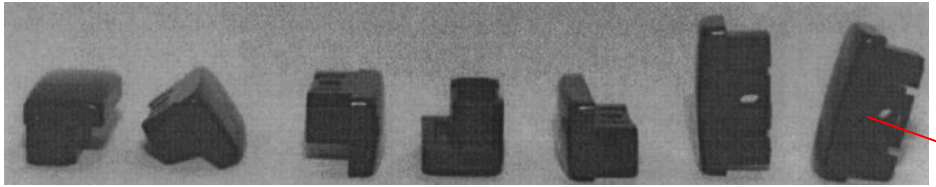
2. The main motivation behind ‘assembly by disassembly’ is to reduce the search space. (T / F)

Part Pose Statistics: Estimators and Experiments

Represents : ‘randomly dropped’ Part Pose Estimator
Estimate the probability of each face making contact with the ground

Intuition :
The most stable pose of the part is the one
in which the part is most likely to land when dropped

Method :
The larger the spherical surface area projected through each face from the component's center of mass, the more stable the component is.



Photographs of the rectangular black stereo button in its **seven stable states**.

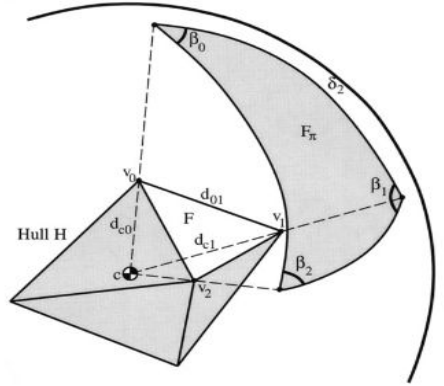


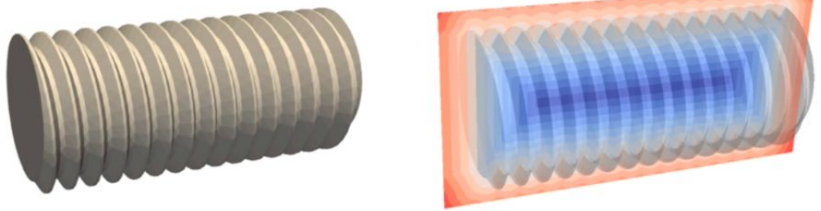
Fig. 3. Computing initial probabilities for each face.

TABLE III
RECTANGULAR BLACK STEREO BUTTON DATA

Pose	Quasi-Static	P'turbed Q–S	Dynamic Sim.	Physical Tests[a]
1	36.2	47.3	54.1	56.0
2	16.0	25.5	24.1	24.5
3	17.4	17.0	14.0	13.6
4	8.1	1.2	1.4	4.4
5	10.6	4.5	5.3	1.4
6	7.5	4.4	1.0	0.3
7	4.3	0.0	0.3	0.0
error	14.0	5.8	1.4	–

KAIST

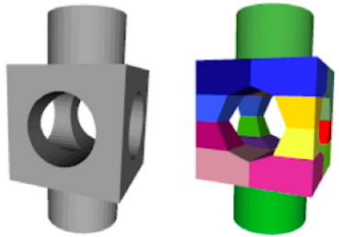
Physics-based Simulation (SDF)



Rigid body dynamics from RedMax [Xu et al. 2021] [Wang et al. 2019]

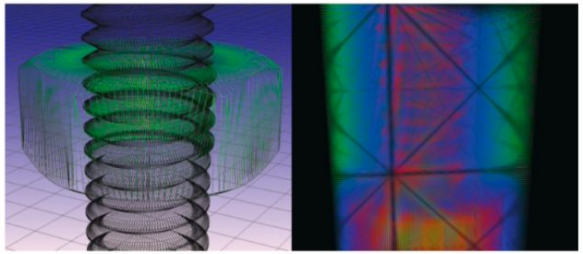
Signed distance field (SDF) for object representation

Accurate and efficient contact-rich simulation (also differentiable and open-source)



Bullet [Coumans and Bai 2021]

Convex decomposition



Factory [Narang et al. 2022] (Concurrent)

Part Pose Statistics: Estimators and Experiments

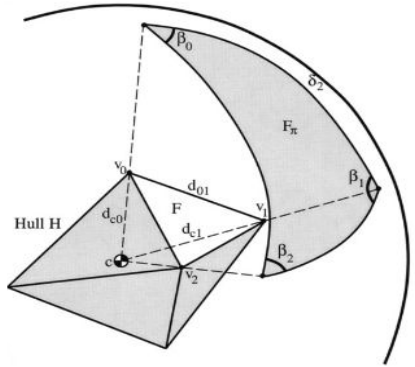
Represents : ‘randomly dropped’ Part Pose Estimator
Estimate the probability of each face making contact with the ground

intuition :
The most stable pose of the part is the pose that is most likely to land when dropped

e.g. thrown vans always land upside (most stable pose)

method :
estimate the probability of the landing with specific face of part by projecting the convex hull into unit sphere, calculate the projected area

probability A_i that the part will land on face F_i under quasi-static conditions.



$$A_i = \frac{\beta_0 + \beta_1 + \beta_2 - \pi}{4\pi}$$

A_i	Projected area (solid angle) of face F_i on the unit sphere
$\beta_0, \beta_1, \beta_2$	Spherical interior angles of the projected triangle

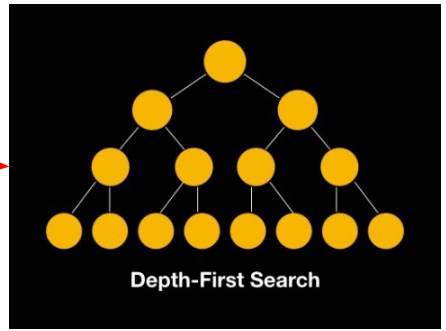
KAIST

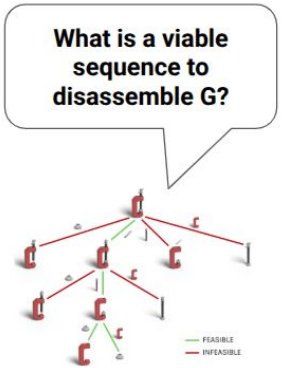
Disassembly Tree Search

Q. “What is a viable sequence to disassemble G?”

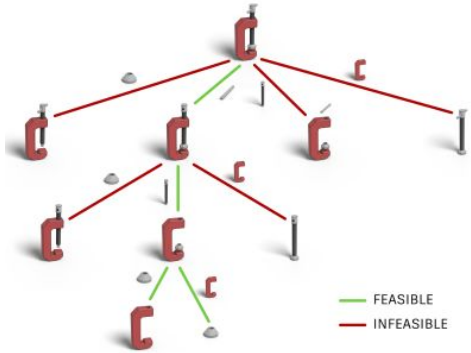
```
def asap():
    tree.add_node(root_node)
    for node in select_node(tree):
        for part in select_part(node):
            for pose in select_pose(node):
                check_assemblable(node, part, pose)
                check_stable(node, part, pose)
                if success:
                    child_node = node \ {part}
                    tree.add_edge(node, child_node)
    return tree
```

Node selection





Finding a feasible method **fast**



KAIST

Tian et al., ASAP: Automated Sequence Planning for Complex Robotic Assembly with Physical Feasibility. ICRA, 2024

Disassembly Tree Search

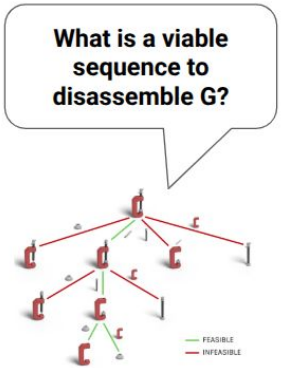
Q. “What is a viable sequence to disassemble G?”

```
def asap():
    tree.add_node(root_node)
    for node in select_node(tree):
        for part in select_part(node):
            for pose in select_pose(node):
                check_assemblable(node, part, pose)
                check_stable(node, part, pose)
                if success:
                    child_node = node \ {part}
                    tree.add_edge(node, child_node)
    return tree
```

Node selection

Part selection

Geometric Heuristic



insight : exterior parts are generally easier to disassemble due to fewer precedence constraint

method : calculate distance between part’s center of mass and assembly center

Parts farther from the assembly center are prioritized for disassembly

Tian et al., ASAP: Automated Sequence Planning for Complex Robotic Assembly with Physical Feasibility. ICRA, 2024