# Differentiable Physics and Stable Modes for Tool Use and Manipulation Planning

Presenter: Steven Patrick

11/9/21

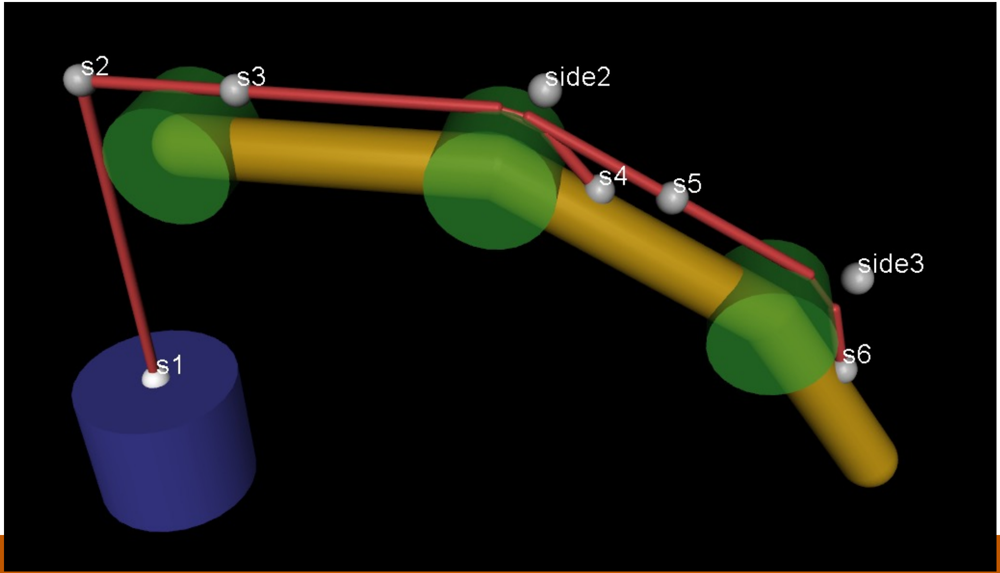
# Task and Manipulation Planning (TAMP)

- The act of planning multiple tasks interacting with the environment to accomplish a goal

- Examples:
  - Cleaning a room
  - Putting furniture together
  - Preparing food

# Differentiable Physics Engines

- If physics are fully differentiable, then “in-theory” you could start at the end position and back-propagate your way to the initial condition and known control inputs.



# Differentiable Physics Engines

- Not enough computation to do this or time steps would be too coarse
- Multiple solutions result in the same final configuration
- Therefore, optimization approaches are promising since most rely heavily on gradient information

# Optimization

$$
\begin{aligned}
\min_u J(u) :=\ & K(x(T)) + \int_0^T L\big(t, x(t), u(t)\big)dt \\
s.t.\ & x(t) \in X, \forall t\ \in [0, T] \\
& x(0) = x_0, \qquad x(t + \Delta t) = f(x, u, \Delta t) \\
& u(t) \in U, \forall t\ \in [0, T] \\
& g_{i(x(t))} \leq 0\ \forall i\ \in [0..L], \forall t\ \in [0, T] \\
& h_{i(x(t))} =\ 0\ \forall\ i \in\ [0..L], \forall t\ \in [0, T]
\end{aligned}
$$

# Optimization

$$\min_u J(u) := K(x(T)) + \int_0^T L\big(t, x(t), u(t)\big)dt$$

$$s.t.\ \ x(t) \in X, \forall t \ \in [0, T]$$

$$x(0) = x_0, \qquad x(t + \Delta t) = f(x, u, \Delta t)$$

$$u(t) \in U, \forall t \ \in [0, T]$$

$$g_{i(x(t))} \leq 0 \ \forall i \ \in [0..L], \forall t \ \in [0, T]$$

$$h_{i(x(t))} = \ 0 \ \forall \ i \in \ [0..L], \forall t \ \in [0, T]$$

Terminal Cost

# Optimization

$$\min_u J(u) := K(x(T)) + \int_0^T L(t, x(t), u(t))\, dt$$

$$s.t.\ x(t) \in X, \forall t \in [0, T]$$

$$x(0) = x_0, \qquad x(t + \Delta t) = f(x, u, \Delta t)$$

$$u(t) \in U, \forall t \in [0, T]$$

$$g_{i(x(t))} \leq 0\ \forall i \in [0..L], \forall t \in [0, T]$$

$$h_{i(x(t))} = 0\ \forall i \in [0..L], \forall t \in [0, T]$$

Running Cost

# Optimization

$$\min_{u} J(u) := K(x(T)) + \int_0^T L\big(t, x(t), u(t)\big)dt$$

$$s.t.\ x(t) \in X, \forall t \in [0, T]$$

$$x(0) = x_0, \qquad x(t + \Delta t) = f(x, u, \Delta t)$$

$$u(t) \in U, \forall t \in [0, T]$$

$$g_{i(x(t))} \leq 0\ \forall i \in [0..L], \forall t \in [0, T]$$

$$h_{i(x(t))} = 0\ \forall\ i \in [0..L], \forall t \in [0, T]$$

State Constraints

# Optimization

$$\min_u J(u) := K(x(T)) + \int_0^T L\big(t, x(t), u(t)\big)dt$$

$$s.t.\ x(t) \in X, \forall t \in [0, T]$$

$$x(0) = x_0, \qquad x(t + \Delta t) = f(x, u, \Delta t)$$

$$u(t) \in U, \forall t \in [0, T]$$

$$g_{i(x(t))} \leq 0\ \forall i \in [0..L], \forall t \in [0, T]$$

$$h_{i(x(t))} = 0\ \forall\ i \in [0..L], \forall t \in [0, T]$$

Initial Condition

# Optimization

$$\min_u J(u) := K(x(T)) + \int_0^T L\big(t, x(t), u(t)\big)dt$$

$$s.t.\ x(t) \in X, \forall t \in [0, T]$$

$$x(0) = x_0, \quad x(t + \Delta t) = f(x, u, \Delta t)$$

$$u(t) \in U, \forall t \in [0, T]$$

$$g_{i(x(t))} \leq 0\ \forall i \in [0..L], \forall t \in [0, T]$$

$$h_{i(x(t))} = 0\ \forall\ i \in [0..L], \forall t \in [0, T]$$

Dynamics Constraints

# Optimization

$$\min_u J(u) := K(x(T)) + \int_0^T L\big(t, x(t), u(t)\big)dt$$

$$s.t.\ x(t) \in X, \forall t \ \in [0, T]$$

$$x(0) = x_0, \qquad x(t + \Delta t) = f(x, u, \Delta t)$$

$$u(t) \in U, \forall t \ \in [0, T]$$

$$g_{i(x(t))} \leq 0 \ \forall i \ \in [0..L], \forall t \ \in [0, T]$$

$$h_{i(x(t))} = \ 0 \ \forall \ i \in \ [0..L], \forall t \ \in [0, T]$$

Control Constraints

# Optimization

$$\min_u J(u) := K(x(T)) + \int_0^T L\big(t, x(t), u(t)\big)dt$$

$$s.t.\ x(t) \in X, \forall t \in [0, T]$$

$$x(0) = x_0, \qquad x(t + \Delta t) = f(x, u, \Delta t)$$

$$u(t) \in U, \forall t \in [0, T]$$

$$g_{i(x(t))} \leq 0\ \forall i \in [0..L], \forall t \in [0, T]$$

$$h_{i(x(t))} = 0\ \forall i \in [0..L], \forall t \in [0, T]$$

Inequality Constraints

# Optimization

$$\min_u J(u) := K(x(T)) + \int_0^T L\big(t, x(t), u(t)\big)dt$$

$$s.t.\ x(t) \in X, \forall t \in [0, T]$$

$$x(0) = x_0, \qquad x(t + \Delta t) = f(x, u, \Delta t)$$

$$u(t) \in U, \forall t \in [0, T]$$

$$g_{i(x(t))} \leq 0\ \forall i \in [0..L], \forall t \in [0, T]$$

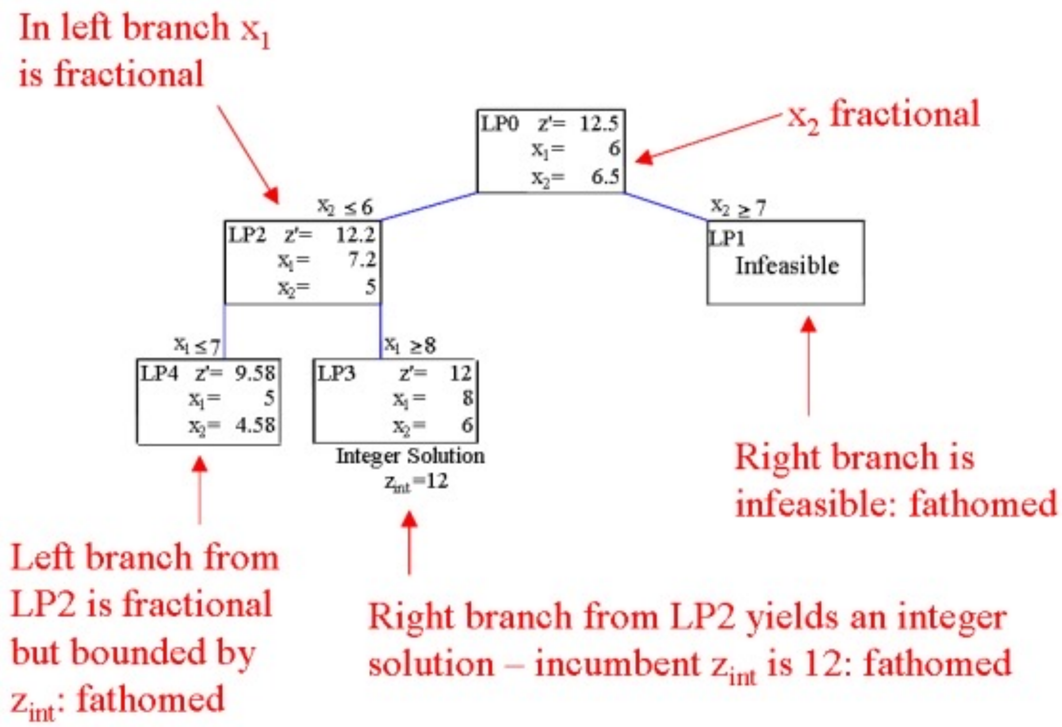
$$h_{i(x(t))} = 0\ \forall\, i \in [0..L], \forall t \in [0, T]$$

Equality Constraints

# Mixed-Integer Programming

- Some or all the optimization variables must have integer values.
- Typically solved with **Branch and Bound** techniques using constraint relaxation.

1. Solve Non-integer problem
2. Branch result into two cases with most "non-integer" value set above and below the nearest integers of the solution
3. Continue branching until all values are integers or no feasible solution

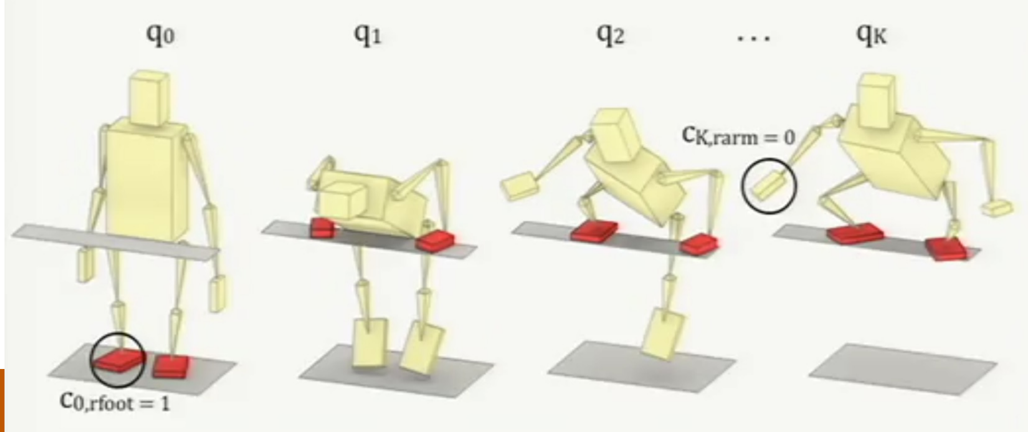


# Contact Invariant Optimization

- Signify contacting a surface with a binary variable
- Add a cost function that penalizes forces when contact is not active

$$L'(x) = L(x) + \sum_i \frac{\|f_i\|^2}{c_i + \epsilon}$$

- $c_i$ is the contact variable, $f_i$ is the force being applied, $\epsilon \sim 10^{-6}$




[*] Discovery of complex behaviors through contact-invariant optimization

# Logic-Geometric Program

$$\min_u J(u) := K\big(x(T)\big) + \int_0^T L\big(\bar{x}(t)\big)dt$$

$$s.t.\ x(0) = x_0, h_{goal}\big(x(T)\big) = 0, g_{goal}\big(x(T)\big) \leq 0$$

$$\forall t \in [0, T]: h_{path}\big(\bar{x}(t), s_{k(t)}\big) = 0$$

$$g_{path}\big(\bar{x}(t), s_{k(t)}\big) \leq 0$$

$$\forall k \in [1, .., K]: h_{switch}(\hat{x}(t_k), a_k) = 0$$

$$g_{switch}(\hat{x}(t_k), a_k) \leq 0$$

$$s_k = f(s_{k-1}, a_k)$$

$\bar{x} = (x, \dot{x}, \ddot{x})$

$\hat{x} = (x, \dot{x}, \dot{x}_{new})$

$a_{1:K} \Rightarrow$ skeleton

# Multi-Bound Tree Search

- 3 Optimization Steps
  - P1: Evaluates initial and final pose without  considering costs
  - P2: Two time-step evaluations per action operator.
  - P3: Fine time-step evaluations over entire trajectory.

# Path Constraint Predicates

| | |
|---|---|
| (touch X Y) | distance between X and Y equal 0 |
| [impulse X Y] | ImpulseExchange eq & skip smoothness constraints on X Y |
| (staFree X Y) | create stable (constrained to zero velocity) free (7D) joint from X to Y |
| (staOn X Y) | create stable 3D $xy\phi$ joint from X to Y |
| (dynFree X) | create dynamic (constrained to gravitational inertial motion) free joint from world to X |
| (dynOn X Y) | create dynamic 3D $xy\phi$ joint from X to Y |
| (inside X Y) | point X is inside object Y $\rightarrow$ inequalities |
| (above X Y) | Y supports X to not fall $\rightarrow$ inequalities |
| (push X Y Z) | (see text) |

$$\forall t \in [0, T]: h_{path}\left(\bar{x}(t), s_{k(t)}\right) = 0$$
$$g_{path}\left(\bar{x}(t), s_{k(t)}\right) \leq 0$$

# Path Constraint Predicates

## Resting and Stable Relations

| | |
|---|---|
| (staFree X Y) | create stable (constrained to zero velocity) free (7D) joint from X to Y |
| (staOn X Y) | create stable 3D $xy\phi$ joint from X to Y |

# Path Constraint Predicates

## Inertial Motion

| | |
|---|---|
| (dynFree X) | create dynamic (constrained to gravitational inertial motion) free joint from world to X |
| (dynOn X Y) | create dynamic 3D $xy\phi$ joint from X to Y |

# Path Constraint Predicates

## Impulse Exchange

| | |
|---|---|
| [impulse X Y] | ImpulseExchange eq & skip smoothness constraints on X Y |

# Path Constraint Predicates

## Geometric

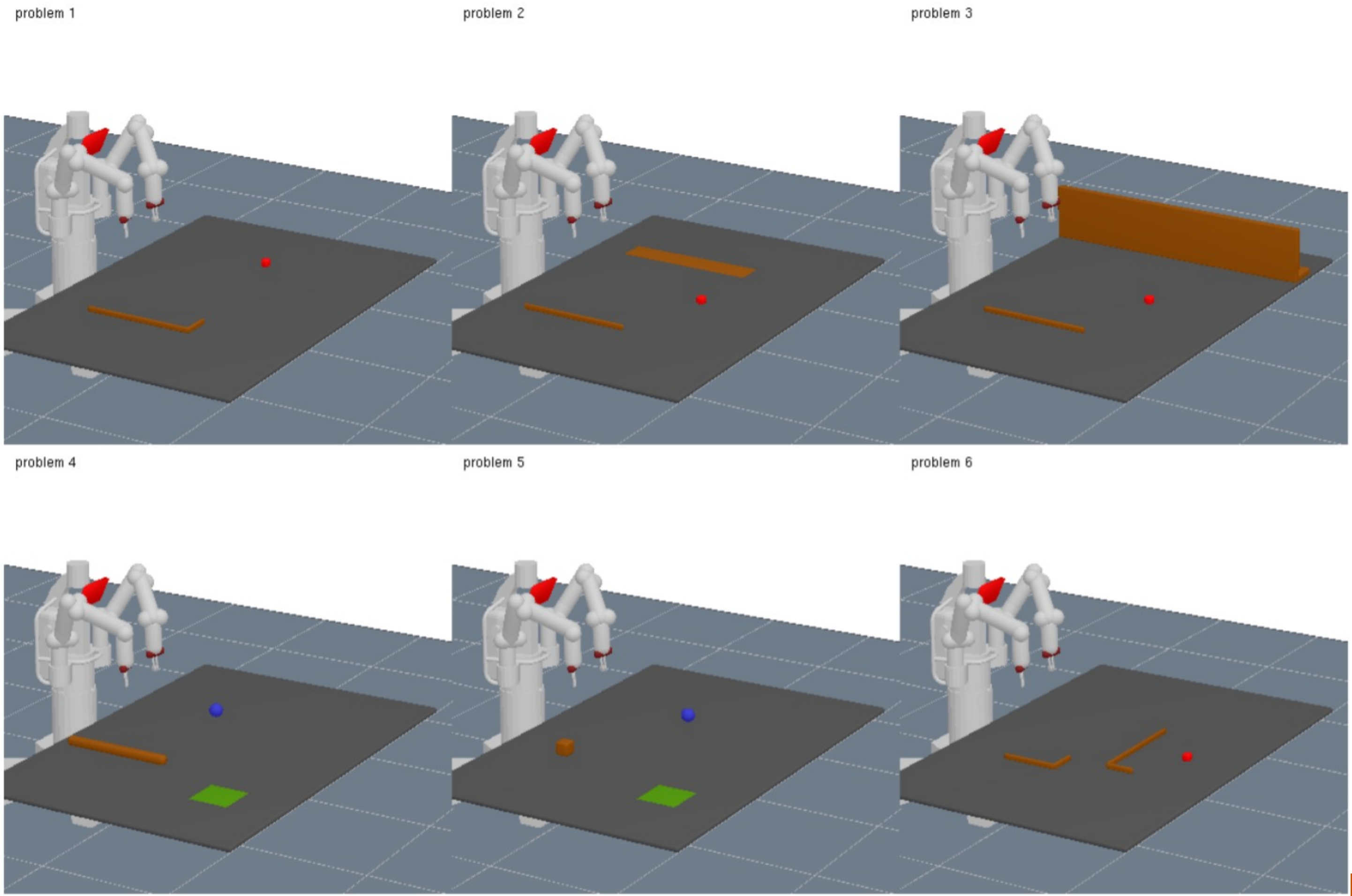
| | |
|---|---|
| (touch X Y) | distance between X and Y equal 0 |
| (inside X Y) | point X is inside object Y $\rightarrow$ inequalities |
| (above X Y) | Y supports X to not fall $\rightarrow$ inequalities |
| (push X Y Z) | (see text) |

# Action Operators

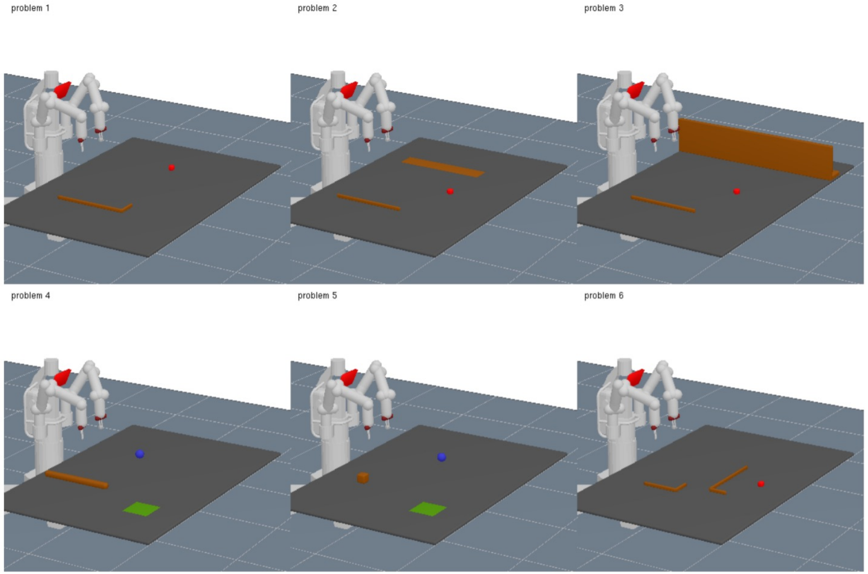
| | |
|---|---|
| grasp(X Y) | [inside X Y] (staFree X Y) |
| handover(X Y Z) | [inside Z Y] (staFree Z Y) |
| place(X Y Z) | [above Y Z] (staOn Z Y) |
| throw(X Y) | (dynFree Y) |
| hit(X Y) | [touch X Y] [impulse X Y] (dynFree Y) |
| hitSlide(X Y Z) | [touch X Y] [impulse X Y] (above Y Z) (dynOn Y Z) |
| hitSlideSit(X Y Z) | "hitSlide(X Y Z)" "place(X Z)" |
| push(X, Y, Z) | komo(push X Y Z) |

$$\forall k \in [1, .., K]: h_{switch}(\hat{x}(t_k), a_k) = 0$$
$$g_{switch}(\hat{x}(t_k), a_k) \leq 0$$

# Results



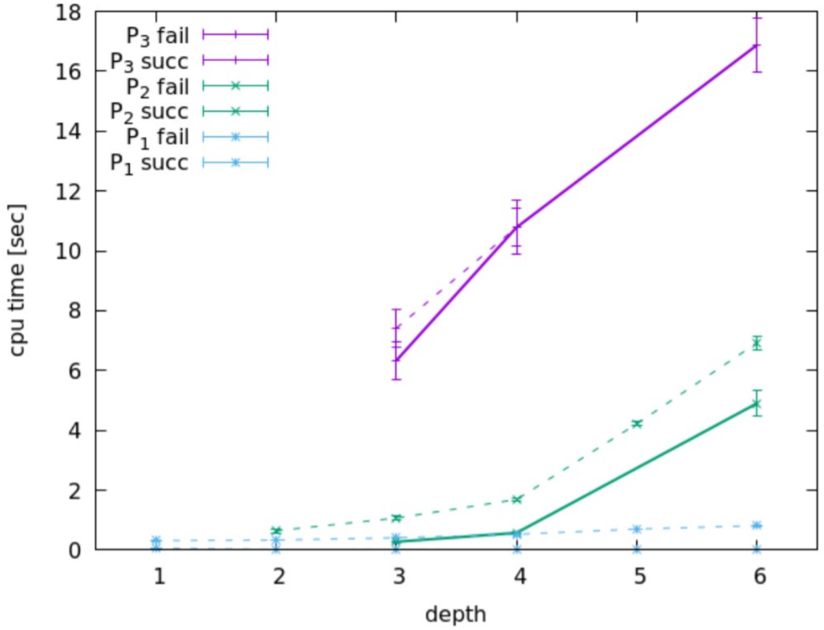
# Results

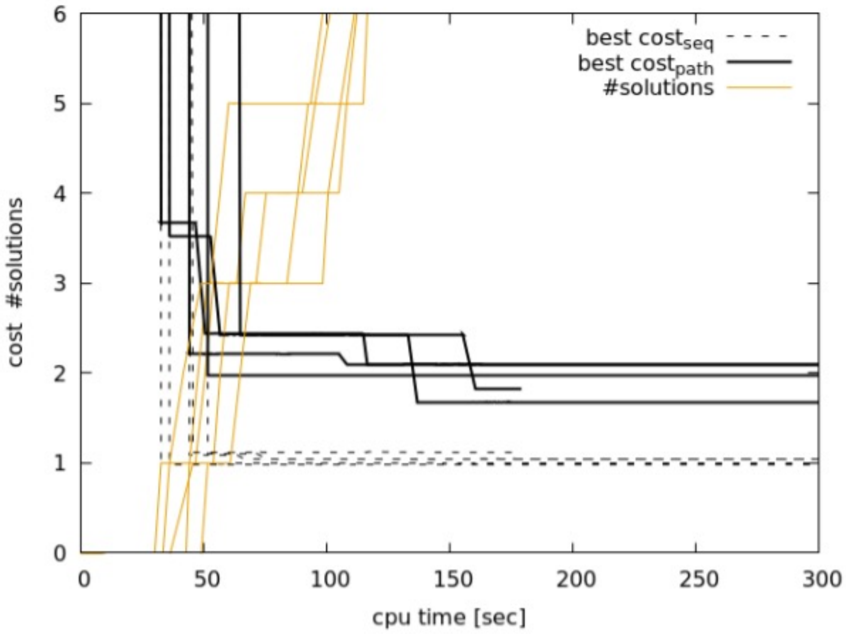



| problem | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| tree size | 12916 | 34564 | 7312 | 12242 | 12242 | 3386 |
| branching | 10.66 | 13.63 | 9.25 | 10.52 | 10.52 | 7.63 |

# Results


# Results (Computation Time)



# Critiques

- Not real-time. Takes a minute or two to generate a trajectory
- Have to hard code knowledge base
- Finds the optimal solution which is usually the least robust

# Conclusions

- This is meant as a new method to explore rather than a total replacement of current manipulation planners
- Has a lot of potential if algorithms are created to reduce computation times and add robustness to solutions

# References

[1] Differentiable Physics and Stable Modes for Tool-Use and Manipulation Planning
[2] Multi-Bound Tree Search for Logic-Geometric Programming in Cooperative Manipulation Domains
[3] Logic-Geometric Programming: An Optimization-Based Approach to Combined Task and Motion Planning
[4] Integrated Task and Motion Planning
[5] Discovery of complex behaviors through contact-invariant optimization
[6] MuJoCo.org

Thank You!

# QUESTIONS?