Conflict-Directed Diverse Planning for Logic-Geometric Programming

Joaquim Ortiz-Haro[1], Erez Karpas[2], Marc Toussaint[1], Michael Katz[3]

[1]TU Berlin,
[2]Technion — Israel Institute of Technology,
[3]IBM Research, Yorktown Heights, NY, USA
{joaquim.ortizdeharo,toussaint}@tu-berlin.de, karpase@technion.ac.il, Michael.Katz1@ibm.com

The 32nd International Conference on Automated Planning and Scheduling

Introduction

Background

Iterative Logical Planning for LGP

Metareasoning for Conflict Extraction

Diverse Logical Planning for LGP

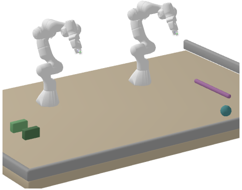
Experimental Results

Conclusion

Introduction I

Combine task planning (what? e.g. "Pick") with motion planning (how? e.g. "Collision Free motion")

Initial State



Symbolic Goal

```
Dark green block on light green block
Blue ball on dark green block
```

Symbolic plans are only a necessary condition. **They often fail in the geometric level!**
E.g. valid symbolic action "(pick blue-ball with left-robot)" fails.
Calls to the motion optimizer are expensive!

Introduction II

- Combine state-of-the-art PDDL Planning with nonlinear optimization (TAMP as Logic Geometric Program)
- Identify and encode geometric infeasibility – **Prefix Conflicts**

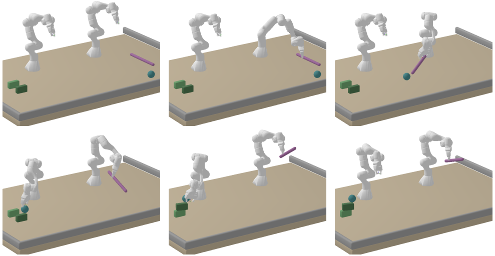


Figure: Manipulation task solved by our algorithm. Solution requires combined logic and geometric reasoning about tool-use, pushing and pick & place actions with two robot manipulators.

Introduction III

Contributions

- Prefixes as geometric conflicts
- Prefix forbidding reformulation
- Metareasoning for conflict extraction
- Diversity criteria for plan selection

Strengths of our method:

- Applicable in any problem in task and motion planning
- Easy implementation with off-the-shelf components:
  - NLP Solver
  - PDDL-Solver

Related Work - TAMP

Task and Motion Planning (TAMP)
Sample-based approaches

- Planner-independent interface layer between task and motion planners [Srivastava et al. (2014)].
- Constraint-based task planning formulation and a Satisfiability Modulo theory (SMT) solver. [Dantam et al. (2016)].
- Constrained samplers (in configuration space) and PDDL planning [Garrett et al. (2020)].
- Precompilation of geometric solutions [Ferrer-Mestres et al. (2017)].

Optimization-based approaches

- Logic Geometric Program formulation [Toussaint et al. (2018)].
- Multi Bound Tree Search solver [Toussaint and Lopes (2017)].

Introduction

Background

Iterative Logical Planning for LGP

Metareasoning for Conflict Extraction

Diverse Logical Planning for LGP

Experimental Results

Conclusion

Background - Classical Planning

Classical Planning task $\Pi = \langle \mathcal{V}, \mathcal{A}, s_0, g, cost \rangle$ ($\textsc{sas}^+$ [Bäckström and Nebel (1995)]).

- $\mathcal{V}$ is a set of *state variables* $v \in \mathcal{V}$, each has a finite domain $\mathcal{D}(v)$. A (partial) state s is a (partial) assignment to the state variables.
- $\mathcal{A}$ is a finite set of *actions*. $a \in \mathcal{A}$ is a pair of of partial states $\langle pre(a), eff(a) \rangle$ called *preconditions* and *effects*. $s[a]$ is the state that results from applying s to a.
- s_0 is the initial state.
- g is the goal (partial state).

$\pi = \langle a_1 \ldots a_K \rangle = a_{1:K}$ is a valid plan if each action is applicable in the previous state ($s_k = s_{k-1}[a_k]$, starting from s_0), and the final state satisfies the goal $g \subseteq s_K$.

Background - Logic Geometric Program

Logic Geometric Program: Joint optimization of logical decision variables $\langle a_1 \dots a_K \rangle$, $\langle s_1 \dots s_K \rangle$ and continuous decision variables $x(t) : t \in \mathbb{R} \to \mathbb{R}^n$ (a trajectory in configuration space).

$$
\begin{aligned}
\textbf{LGP} \quad & \min_{x, s_{1:K}, a_{1:K}, K} \sum_{k=0}^{K-1} \int_{kT}^{(k+1)T} c(x(t), s_{0:k}) \; dt && \text{(1a)} \\
& x(0) = x_0, \\
& \forall k \in 0, \dots, K : \\
& \quad h_k(x(t), s_{0:k}) \leq 0, \;\; t \in [kT, (k+1)T], \\
& \quad s_k = s_{k-1}[a_k], \\
& g \subseteq s_K.
\end{aligned}
$$

Logic: SAS+ Planning task $\Pi = \langle \mathcal{V}, \mathcal{A}, s_0, g, cost \rangle$.
Nonlinear constraints $h(\cdot)$ and cost $c(\cdot)$ on the trajectory.

Background - Logic Geometric Program II

A sequence of logical states and actions imply a nonlinear program (NLP) on the trajectory.

$$\mathbf{NLP}(a_{1:K}): \quad \min_x \sum_{k=0}^{K-1} \int_{kT}^{(k+1)T} c(x(t), s_{0:k})\ dt \tag{2a}$$

$$\text{s.t. } x(0) = x_0, \tag{2b}$$

$$\forall k \in 0, \dots, K: \tag{2c}$$

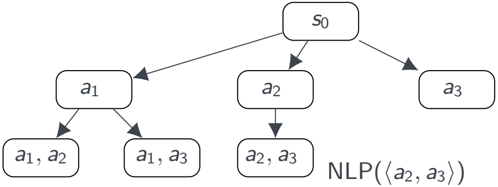
$$h_k(x(t), s_{0:k}) \leq 0,\ \ t \in [kT, (k+1)T].$$

where $s_{1:K}$ is uniquely defined by $a_{1:K}$ and s_0.
A sequence of actions $a_{1:K}$ is geometrically infeasible when $\mathbf{NLP}(a_{1:K})$ is infeasible, i.e

$$\nexists\ x(t),\ \ t \in [0, TK]\ \ \text{Eq. (2b) (2c)}. \tag{3}$$

Background - Multibound Tree Search (MBTS)

Tree search starting from s_0.
Explore $\langle s_0, a_1, s_1, a_2, s_3 \ldots \rangle$ to find nodes $s_n \supseteq g$ with feasible $NLP(\langle a_1 \ldots a_K \rangle)$.
[Toussaint and Lopes (2017)]



Bounds before solving $\mathbf{NLP}(a_{1:K})$: consider only mode-switches $x(t_k)$ $k = 1, ..., K$.

- Pose bound: compute $x(t_k)$ independently.
- Sequence bound: compute $\{x(t_k)\}$ jointly.

Combined Logic-Geometric search:

- Logic expansion using breadth first search.
- Three queues to solve nonlinear programs (Pose, Sequence, Full).

Introduction

Background

Iterative Logical Planning for LGP

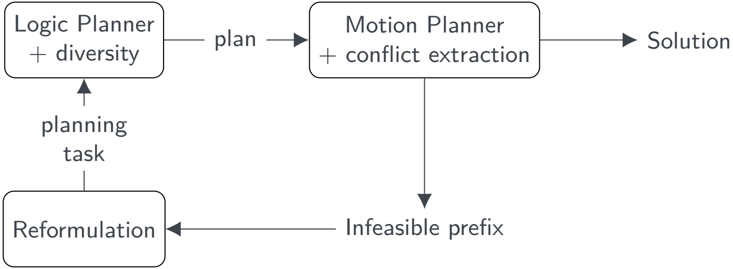
Metareasoning for Conflict Extraction

Diverse Logical Planning for LGP

Experimental Results

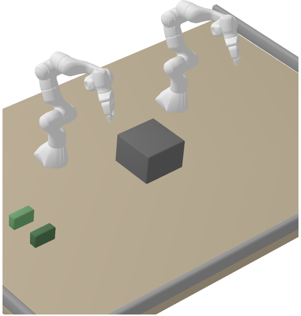
Conclusion

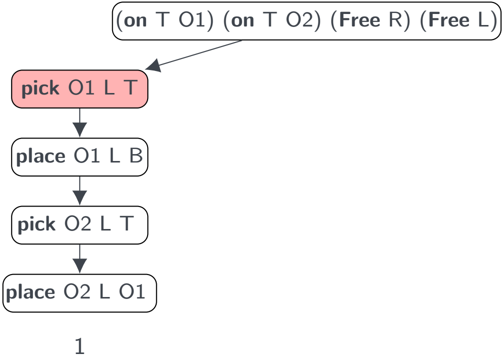
Iterative Logic Planning for LGP



Overview: we combine a symbolic planner that generates plans of actions, with a motion planner to compute a trajectory. If a plan is not geometrically feasible, we extract a conflict, namely a prefix of infeasible actions, and reformulate the planning task.

Illustrative example of the execution of our algorithm (with *N=1* and *eager* conflict extraction). The scene contains two movable objects *O1*, *O2*, a table *T*, a box *B* and two robots *R*, *L* that can *pick* and *place* the objects. The goal is *(on B O1)*, *(on O1 O2)*.





(**on** T O1) (**on** T O2) (**Free** R) (**Free** L)

pick O1 L T

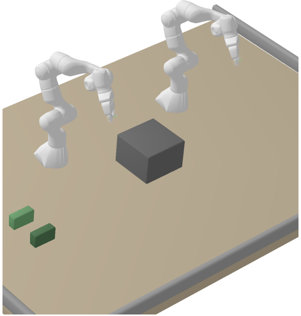
place O1 L B

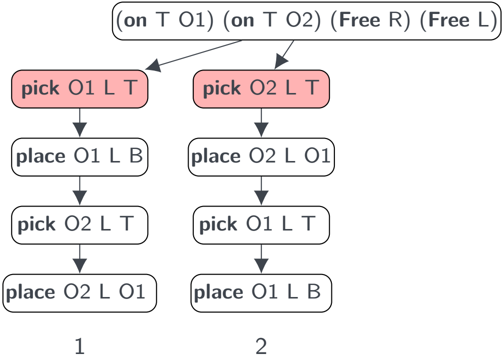
pick O2 L T

place O2 L O1

1

Illustrative example of the execution of our algorithm (with *N=1* and *eager* conflict extraction). The scene contains two movable objects *O1*, *O2*, a table *T*, a box *B* and two robots *R*, *L* that can *pick* and *place* the objects. The goal is *(on B O1)*, *(on O1 O2)*.





(**on** T O1) (**on** T O2) (**Free** R) (**Free** L)

pick O1 L T

place O1 L B

pick O2 L T

place O2 L O1

1

pick O2 L T

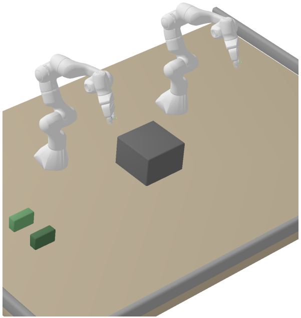
place O2 L O1

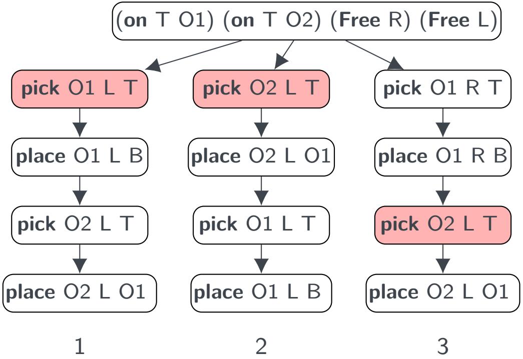
pick O1 L T

place O1 L B

2

Illustrative example of the execution of our algorithm (with *N=1* and *eager* conflict extraction). The scene contains two movable objects *O1*, *O2*, a table *T*, a box *B* and two robots *R*, *L* that can *pick* and *place* the objects. The goal is *(on B O1)*, *(on O1 O2)*.





(**on** T O1) (**on** T O2) (**Free** R) (**Free** L)

1

pick O1 L T

place O1 L B

pick O2 L T

place O2 L O1

2

pick O2 L T

place O2 L O1

pick O1 L T

place O1 L B

3

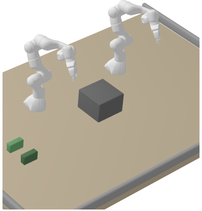
pick O1 R T

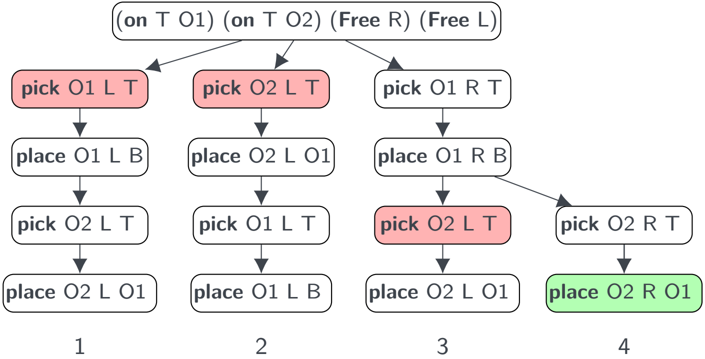
place O1 R B

pick O2 L T

place O2 L O1

Illustrative example of the execution of our algorithm (with *N=1* and *eager* conflict extraction). The scene contains two movable objects *O1*, *O2*, a table *T*, a box *B* and two robots *R*, *L* that can *pick* and *place* the objects. The goal is *(on B O1)*, *(on O1 O2)*.





(**on** T O1) (**on** T O2) (**Free** R) (**Free** L)

pick O1 L T
place O1 L B
pick O2 L T
place O2 L O1
1

pick O2 L T
place O2 L O1
pick O1 L T
place O1 L B
2

pick O1 R T
place O1 R B
pick O2 L T
place O2 L O1
3

pick O2 R T
place O2 R O1
4

Prefixes as Conflicts

Theorem: Let Π be an LGP task (1), and let π be a sequence of logical actions, such that π is not geometrically feasible from the initial state. Then any sequence of actions π' which contains π as a prefix is not geometrically feasible from the initial state.

Example:
If $\langle \text{pick}(B) \rangle$ is infeasible at the geometric level $\rightarrow$ $\langle \text{pick}(B), \text{place}(B) \rangle$ is infeasible.

However, it is *not* safe to infer that $\langle \text{pick}(B), \text{place}(B) \rangle$ can never be applied. The sequence of actions $\langle \text{pick}(A), \text{place}(A), \text{pick}(B), \text{place}(B) \rangle$ might be geometrically feasible.

Forbidding Plans by prefixes

Prefix forbidding reformulation, based on [Katz et al. (2018)].

- Forbid $\langle a_1 \dots a_K \rangle$ as a prefix, instead of as a plan. Applying the starting sequence $\langle a_1 \dots a_K \rangle$ in the reformulated task leads to a dead end.
- Simultaneously forbid multiple prefixes (compact). Build a *prefix tree* $T = (N, E)$ which contains all (non-dominated) prefixes. Each node corresponds to a prefix, and there is an edge from node π to node π' if we can add one action to π to yield π'.

Forbidding Plans by prefixes II

Definition: Let $\Pi = \langle \mathcal{V}, \mathcal{A}, s_0, g, cost \rangle$ be a planning task, $T = (N, E)$ be a prefix tree with $L \subseteq N$ being the leaf nodes, and $\mathcal{A}(T)$ be the set of operators that appear on the prefixes in T. The task $\Pi^-_T = \langle \mathcal{V}', \mathcal{A}', s'_0, g', cost' \rangle$ is:

- $\mathcal{V}' = \mathcal{V} \cup \{\overline{v}\} \cup \{\overline{v}_s \mid s \in N\}$, with $\overline{v}_s$ binary and $\overline{v}$ a ternary variable,
- $\mathcal{A}' = \mathcal{A}^e \cup \mathcal{A}^1 \cup \mathcal{A}^2 \cup \mathcal{A}^3$, where
  $\mathcal{A}^e = \{a^e \mid a \in \mathcal{A} \setminus \mathcal{A}(T)\}$, $\mathcal{A}^1 = \{a^1 \mid a \in \mathcal{A}\}$, $\mathcal{A}^2 = \{a^2 \mid a \in \mathcal{A}(T)\}$, and
  $\mathcal{A}^3 = \{a^3_{(s,t)} \mid (s,t) \in E\}$ with

$$\begin{aligned}
a^e &= \langle pre(a) \cup \{\langle \overline{v}, 1\rangle\}, eff(a) \cup \{\langle \overline{v}, 0\rangle\}\rangle \\
a^1 &= \langle pre(a) \cup \{\langle \overline{v}, 0\rangle\}, eff(a)\rangle \\
a^2 &= \langle pre(a) \cup \{\langle \overline{v}, 1\rangle\} \cup \{\langle \overline{v}_s, 0\rangle \mid (s,t) \in E^a\}, eff(a) \cup \{\langle \overline{v}, 0\rangle\}\rangle \\
a^3_{(s,t)} &= \langle pre(a_{(s,t)}) \cup \{\langle \overline{v}, 1\rangle, \langle \overline{v}_s, 1\rangle\}, eff(a_{(s,t)}) \cup \{\langle \overline{v}_s, 0\rangle, \langle \overline{v}_t, 1\rangle\}\rangle \text{ if } t \notin L, \\
a^3_{(s,t)} &= \langle pre(a_{(s,t)}) \cup \{\langle \overline{v}, 1\rangle, \langle \overline{v}_s, 1\rangle\}, \{\langle \overline{v}, 2\rangle\}\rangle \text{ if } t \in L,
\end{aligned}$$

- $s_0'[v] = s_0[v]$ for all $v \in \mathcal{V}$, $s_0'[\overline{v}] = 1$, $s_0'[\overline{v}_{s_0}] = 1$, and $s_0'[\overline{v}_s] = 0$ for all $s \in N \setminus \{s_0\}$, and
- $g'[v] = g[v]$ for all $v \in \mathcal{V}$ s.t. $g[v]$ defined, and $g'[\overline{v}] = 0$.

Feasibilty Checking

Let $\langle a_1 \dots a_K \rangle$ be an infeasible plan. How to compute an infeasible prefix?

- Lazy: Return $\langle a_1 \dots a_K \rangle$
- Eager: Find the mininal infeasible prefix

$$\min \; k \text{ s.t } \mathrm{Feas}(\langle a_1 \dots a_k \rangle) = 0 \quad \text{(i.e. } \langle a_1 \dots a_k \rangle \text{ is infeasible)} \tag{4}$$

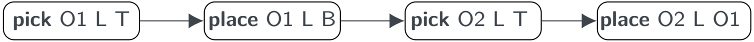
Binary Search: $\mathrm{Feas}(\langle a_1 \dots a_k \rangle) \geq \mathrm{Feas}(\langle a_1 \dots a_{k+1} \rangle)$. Each check → solve $\textbf{NLP}(a_{1:k})$

Goal: Reduce the global number of calls to the nonlinear optimizer.
Trade-off: finding short prefixes requires solving more NLPs now, but less in the future.

Store all solved NLPs in a Cache

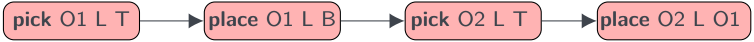
Feasibility Checking – Example

The logic planner computes the plan $\langle a_1, a_2, a_3, a_4 \rangle$



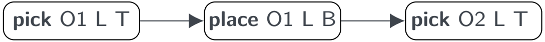
pick O1 L T → **place** O1 L B → **pick** O2 L T → **place** O2 L O1

NLP$(a_{1:4})$ is infeasible



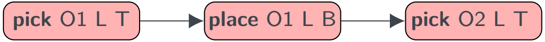
pick O1 L T → **place** O1 L B → **pick** O2 L T → **place** O2 L O1

We can return $\langle a_1, a_2, a_3, a_4 \rangle$ as a conflict, or look for a smaller prefix that is infeasible.
For example, we can evaluate $\langle a_1, a_2, a_3 \rangle$



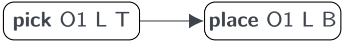
pick O1 L T → **place** O1 L B → **pick** O2 L T

NLP$(a_{1:3})$ is also infeasible



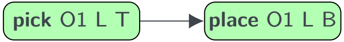
pick O1 L T → **place** O1 L B → **pick** O2 L T

We now evaluate $\langle a_1, a_2 \rangle$



pick O1 L T → **place** O1 L B

NLP$(a_{1:2})$ is feasible – We return $\langle a_1, a_2, a_3 \rangle$ as conflict.



pick O1 L T → **place** O1 L B

Introduction

Background

Iterative Logical Planning for LGP

Metareasoning for Conflict Extraction

Diverse Logical Planning for LGP

Experimental Results

Conclusion

Metareasoning for Conflict Extraction

- Middle-ground approach between lazy and eager.
- When searching for the conflict in $a_{1:K}$, we have a range $[l, u]$ such that $a_{1:u}$ is not geometrically feasible, while $a_{1:l}$ is. We can decide to return $a_{1:u}$ as conflict or search a smaller conflict (between l, u).

Markov Decision Process (MDP)

- States $S_\pi = \{\langle l, u\rangle \mid l \leq u \in 0 \dots K\}$ – describe the current range of the search. Terminal: converged search $\langle u, u\rangle$. Start: $\langle 0, K\rangle$
- Actions at $\langle l, u\rangle$: A) stop searching (reach state $\langle u, u\rangle$), or B) check any node $l < m < u$. We reach state $\langle m, u\rangle$ with probability $p_f(a_{1:m})$, and state $\langle l, m\rangle$ with probability $1 - p_f(a_{1:m})$.
- $r(\langle u, u\rangle) = r(\tau)$ rewards from adding the conflict $\tau = \langle a_1 \dots a_u\rangle$

Metareasoning for Conflict Extraction

- How to estimate reward?

  $\hat{r}(\tau) = \frac{|\{\pi' | \pi' \in C, \tau \text{ is a prefix of } \pi'\}|}{|C|}$, C is the set of prefixes in our cache.

- How to estimate the transition probabilities?

  Use prefix length n as feature. Ratio between feasible and infeasible prefixes of given length. $p(n) = \frac{\# \text{ feasible Prefixes}(n)}{\#\text{evaluated Prefixes } (n)}$.

Metareasoning based conflict extraction
Solve MDP with Dynamic Programming (backwards recursion).
At current $\langle l, u \rangle$ we choose the optimal action (i.e. search an intermediate node or stop) in the MDP.

Introduction

Background

Iterative Logical Planning for LGP

Metareasoning for Conflict Extraction

Diverse Logical Planning for LGP

Experimental Results

Conclusion

Diverse Logical Planning for LGP

We can generate a set of plans in each iteration (instead of only one)
How to choose which plan to test next? → **Prefix Novelty**

- High chance to extract a short conflict from that plan.
- Explore the space of logical plans. Different logical plans → different nonlinear programs.

Novelty of a plan π with respect to a set of plans LP:

$$np(\pi, LP) := -\min\{k \mid \forall \pi' \in LP, \pi'|_k \neq \pi|_k\}$$

Choose test plan π that maximizes np

Complete Algorithm

Algorithm 1

```
Input: LGP task $\Pi_{\text{LGP}}$
Parameters: $N$
$\Pi :=$ logical projection of $\Pi_{\text{LGP}}$
$T, LP, MC := \emptyset$                                  ▷ set of tried plans, found plans, found conflicts
while not solved do
    $\Pi^f :=$ FORBID($\Pi, LP \cup MC$)
    $LP := LP \cup$ Diverse-Plan($\Pi^f, N$)
    $\pi :=$ SELECT($LP, T$)
    feasible, traj := MOTION-Feasible?($\Pi_{\text{LGP}}, \pi$)
    if feasible then return $\pi$, traj
    else
        $T := T \cup \{\pi\}$
        conflict := FIND-CONFLICT($\pi$)
        $MC := MC \cup \{$conflict$\}$
    end if
end while
```

Complete Algorithm II

Theorem: If the underlying classical planner is sound and complete and the motion planner always finds a feasible trajectory if such a trajectory exists, then Algorithm 1 is sound and complete.
Proof: The proof follows from the fact that we only identify prefixes which can not appear in the beginning of geometrically feasible plans, and from the correctness of the forbidding compilation.

Introduction

Background

Iterative Logical Planning for LGP

Metareasoning for Conflict Extraction

Diverse Logical Planning for LGP

Experimental Results

Conclusion

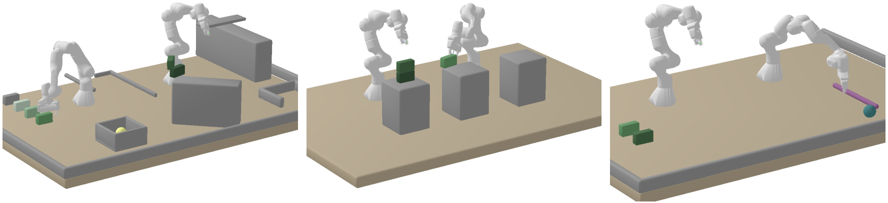
Experimental Results

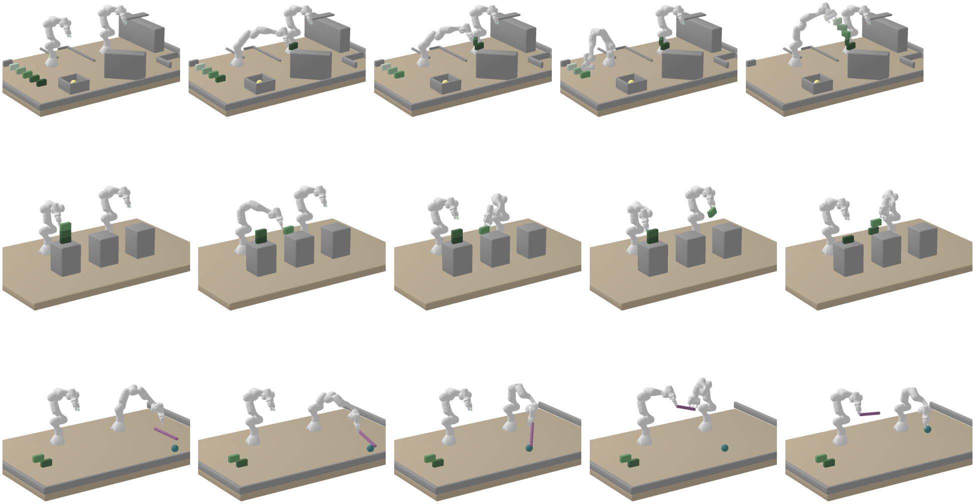
Benchmark Scenarios (Two 7-DOF robots)

Blocks (left) pick and place actions to construct a tower of blocks. Robots can hold a stack of blocks, move the boxes and place several objects on top of other objects.

Hanoi (middle) the robots can execute pick and place actions, to solve a tower of Hanoi problem with objects of equal size and three tables.

Push (right) the robots can pick and place blocks and balls, and pick sticks and use them as a tool to push balls.





LGP Formulation

Logic Minimal set of logical symbols.
Symbolic structure of kinematic tree and interactions (grasp vs pushing).
Examples predicates: *on(A B)*, *busy(gripper)*.
Example actions: *(pick A gripper T)*, *(push A stick table)*.

Geometry Collision avoidance, reachability, physical interactions and placement constraints.
Grasping model: a point near the endeffector's grasp palm touches the object surface. Grasp → stable relative transformation.

Pushing model: decision variables for the force and point-of-attack. Motion and forces are constrained by physics equations [Toussaint et al. (2020)].

Baselines

Variations of Multibound Tree search (*MBTS-{0,1,2}*).

MBTS-0 no geometric checks in intermediate symbolic nodes

MBTS-{1,2} check, respectively, the pose and sequence bound before expanding a node.

Metrics

- Computational Time (time [s])
- Number of Pose Bounds (pose)
- Number of Sequence Bounds (seq)

+ **Abblation Study**

NOTE: The scope of the paper is to improve the logic search in the LGP framework, so that the formulation can be applied to settings that require longer action sequences and challenging symbolic reasoning. Therefore, we do not compare to other methods in task and motion planning, e.g. [Garrett et al. (2020)], that use different underlying problem formulations and methods.

	MBTS-0			N=1 *eager*			N=4 *eager*			N=4 *meta*		
	time	pose	seq	time	pose	seq	time	pose	seq	time	pose	seq
blocks-0	**19.4**$_{\mathbf{1.0}}$	12.0$_{0.0}$	3.0$_{0.0}$	43.7$_{1.8}$	19.9$_{0.9}$	6.5$_{0.5}$	41.8$_{4.3}$	17.5$_{1.9}$	6.5$_{0.9}$	41.3$_{4.4}$	17.5$_{1.9}$	2.8$_{0.6}$
blocks-1	-	-	-	44.8$_{1.3}$	18.0$_{0.0}$	5.0$_{0.0}$	**44.0**$_{\mathbf{5.6}}$	17.1$_{2.3}$	4.8$_{0.9}$	46.5$_{6.4}$	17.1$_{2.3}$	1.7$_{0.3}$
blocks-2	-	-	-	82.6$_{10.5}$	17.0$_{0.0}$	4.0$_{0.0}$	**60.4**$_{\mathbf{8.6}}$	12.6$_{1.1}$	2.4$_{0.5}$	70.9$_{11.8}$	12.6$_{1.1}$	1.4$_{0.2}$
blocks-3	-	-	-	111$_{5.8}$	17.0$_{1.0}$	3.4$_{0.4}$	104$_{19.7}$	21.7$_{2.8}$	5.4$_{1.1}$	**80.2**$_{\mathbf{12.0}}$	20.8$_{3.1}$	2.1$_{0.4}$
blocks-4	-	-	-	200$_{33.1}$	27.1$_{8.2}$	6.1$_{2.8}$	160$_{17.0}$	19.3$_{3.1}$	3.3$_{1.1}$	**139**$_{\mathbf{22.6}}$	17.8$_{2.7}$	1.3$_{0.2}$
hanoi-0	10.4$_{0.4}$	13.0$_{0.0}$	4.0$_{0.0}$	**7.0**$_{\mathbf{0.2}}$	7.0$_{0.0}$	3.0$_{0.0}$	10.0$_{1.9}$	8.0$_{0.9}$	3.7$_{0.8}$	9.1$_{1.8}$	8.6$_{1.1}$	2.9$_{0.5}$
hanoi-1	34.7$_{0.7}$	34.0$_{0.0}$	6.0$_{0.0}$	27.0$_{0.6}$	17.0$_{0.0}$	8.0$_{0.0}$	18.7$_{3.1}$	13.5$_{1.0}$	5.1$_{0.6}$	**13.8**$_{\mathbf{2.2}}$	14.0$_{1.0}$	3.4$_{0.4}$
push-1	41.9$_{0.8}$	55.8$_{0.2}$	1.0$_{0.0}$	**17.1**$_{\mathbf{0.4}}$	14.0$_{0.0}$	4.0$_{0.0}$	24.4$_{1.4}$	17.3$_{1.1}$	5.3$_{0.5}$	24.9$_{1.7}$	18.7$_{1.2}$	3.8$_{0.4}$
push-2	50.0$_{1.0}$	64.0$_{0.0}$	1.0$_{0.0}$	49.5$_{0.9}$	37.0$_{0.0}$	13.2$_{0.1}$	37.1$_{1.1}$	23.2$_{0.9}$	7.2$_{0.4}$	**34.3**$_{\mathbf{1.6}}$	24.3$_{0.8}$	3.2$_{0.2}$
push-3	27.7$_{0.9}$	38.0$_{0.0}$	1.0$_{0.0}$	**14.4**$_{\mathbf{0.2}}$	11.0$_{0.0}$	3.0$_{0.0}$	26.1$_{3.2}$	17.9$_{2.0}$	5.8$_{0.9}$	21.1$_{1.8}$	17.3$_{1.8}$	2.9$_{0.3}$
push-4	75.9$_{1.5}$	104$_{0.0}$	2.0$_{0.0}$	71.3$_{7.8}$	41.2$_{2.8}$	15.2$_{1.5}$	32.6$_{4.1}$	20.3$_{2.2}$	5.9$_{1.0}$	**30.6**$_{\mathbf{2.7}}$	21.3$_{2.4}$	3.1$_{0.4}$
push-5	111$_{1.6}$	144$_{0.1}$	1.0$_{0.0}$	**20.4**$_{\mathbf{0.3}}$	17.0$_{0.0}$	5.0$_{0.0}$	30.8$_{2.5}$	23.7$_{2.2}$	7.4$_{0.9}$	29.4$_{2.4}$	24.4$_{2.3}$	3.2$_{0.4}$
push-6	117$_{1.5}$	142$_{0.0}$	1.0$_{0.0}$	64.5$_{1.2}$	50.0$_{0.0}$	17.1$_{0.1}$	45.4$_{1.0}$	29.2$_{0.7}$	9.2$_{0.4}$	**45.1**$_{\mathbf{1.2}}$	31.8$_{1.2}$	4.6$_{0.3}$
push-7	-	-	-	**68.6**$_{\mathbf{4.6}}$	51.3$_{3.5}$	19.0$_{1.6}$	79.1$_{5.4}$	52.6$_{4.0}$	18.0$_{1.6}$	70.4$_{2.6}$	53.0$_{2.7}$	7.4$_{0.5}$
push-8	78.3$_{1.2}$	92.0$_{0.0}$	1.0$_{0.0}$	**17.0**$_{\mathbf{0.4}}$	13.0$_{0.0}$	3.0$_{0.0}$	32.4$_{3.6}$	26.0$_{3.0}$	7.8$_{1.1}$	32.8$_{3.7}$	28.2$_{3.6}$	4.1$_{0.6}$
push-9	248$_{40.1}$	423$_{67.2}$	2.5$_{0.5}$	63.3$_{1.3}$	45.0$_{0.0}$	16.0$_{0.0}$	**46.2**$_{\mathbf{6.2}}$	32.5$_{3.7}$	10.4$_{1.4}$	49.8$_{11.9}$	39.7$_{9.5}$	5.3$_{1.7}$
push-10	**12.7**$_{\mathbf{0.5}}$	16.0$_{0.0}$	1.0$_{0.0}$	12.8$_{0.5}$	9.0$_{0.0}$	3.0$_{0.0}$	13.8$_{1.6}$	10.4$_{1.3}$	3.3$_{0.6}$	13.2$_{1.4}$	10.5$_{1.3}$	1.6$_{0.2}$
push-11	-	-	-	61.1$_{9.3}$	25.5$_{2.3}$	10.7$_{1.4}$	**26.0**$_{\mathbf{2.0}}$	13.4$_{0.5}$	3.8$_{0.4}$	30.5$_{5.4}$	16.7$_{2.5}$	2.6$_{0.6}$
Total	827	1138	24.5	976	437	145	833	376	115	783	394	57.4

Results I

- **Comparison to baseline** Hypothesis: *"Our basic novel approach (N=1, eager conflict extraction) will be faster and solve more problems than any of the MBTS baselines"*.

	Problems solved	Problems faster
N=1,eager	18	16
MBTS-0	16	2

* MBTS-0 fails in problems with long action sequences/high branching factor.

* Ours: state-of-the-art PDDL-Planner + encoding of geometric information.

Results II

- **Analysis of diverse planning** Hypothesis: *"Diverse planning with novelty measure will improve over incremental plan generation"*.

	time	pose	seq
N=1	976	437	145
N=4	833	376	115

* $N = 4$ reduces computational time and the number geometric checks.

* Prefixes and orderings in LGP are important. Our measure outperforms classical plan similarity metrics like action set similarity.

Results III

- **Analysis of conflict extraction** Hypothesis: *"Metareasoning is faster than eager and lazy conflict extraction"*.

	time	pose	seq
N=4, Metareasoning	783	394	57
N=4, Eager	833	376	115

* For *N=4*, *eager* (finds minimal prefix using the *sequence* bound), *meta* (metareasoning approach for conflict extraction).

* Metareasoning speedup (*meta* is better in 12 vs *eager* 6).

* The Sequence bound of a feasible NLP is very fast to compute.

Introduction

Background

Iterative Logical Planning for LGP

Metareasoning for Conflict Extraction

Diverse Logical Planning for LGP

Experimental Results

Conclusion

Conclusion

- Systematic interface between a PDDL-Planner and a nonlinear solver to solve LGP. Identify and encode geometric conflicts (infeasible prefixes).
- Extension with novelty selection criteria and metareasoning.
- Outperform previous solvers for LGP.

Future Work:

- Combine geometric and logical heuristics (with information about feasible prefixes).
- Detect and encode stronger conflicts (subsets of infeasible constraints instead of prefixes).

Preprint: *J. Ortiz-Haro, E. Karpas, M. Katz and M. Toussaint (2022). A Conflict-driven Interface between Symbolic Planning and Nonlinear Constraint Solving. Under review, submitted to IEEE Robotics and Automation Letters (RA-L).*

Conflict-Directed Diverse Planning for Logic-Geometric Programming

Thanks for you attention!

References I

Bäckström, C. and Nebel, B. (1995). Complexity results for SAS$^+$ planning. *Computational Intelligence*, 11(4):625–655.

Dantam, N. T., Kingston, Z. K., Chaudhuri, S., and Kavraki, L. E. (2016). Incremental task and motion planning: A constraint-based approach. In *Robotics: Science and systems*, volume 12, page 00052. Ann Arbor, MI, USA.

Ferrer-Mestres, J., Francès, G., and Geffner, H. (2017). Combined task and motion planning as classical AI planning. *CoRR*, abs/1706.06927.

Garrett, C. R., Lozano-Pérez, T., and Kaelbling, L. P. (2020). Pddlstream: Integrating symbolic planners and blackbox samplers via optimistic adaptive planning. In *Proceedings of the Thirtieth International Conference on Automated Planning and Scheduling, Nancy, France, October 26-30, 2020*, pages 440–448. AAAI Press.

Katz, M., Sohrabi, S., Udrea, O., and Winterer, D. (2018). A novel iterative approach to top-k planning. In *Proceedings of the Twenty-Eighth International Conference on Automated Planning and Scheduling, ICAPS 2018, Delft, The Netherlands, June 24-29, 2018*, pages 132–140. AAAI Press.

References II

Srivastava, S., Fang, E., Riano, L., Chitnis, R., Russell, S. J., and Abbeel, P. (2014). Combined task and motion planning through an extensible planner-independent interface layer. In *2014 IEEE International Conference on Robotics and Automation, ICRA 2014, Hong Kong, China, May 31 - June 7, 2014*, pages 639–646. IEEE.

Toussaint, M., Allen, K. R., Smith, K. A., and Tenenbaum, J. B. (2018). Differentiable physics and stable modes for tool-use and manipulation planning. In *Robotics: Science and Systems XIV, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA, June 26-30, 2018*.

Toussaint, M., Ha, J., and Driess, D. (2020). Describing physics for physical reasoning: Force-based sequential manipulation planning. *IEEE Robotics Autom. Lett.*, 5(4):6209–6216.

Toussaint, M. and Lopes, M. (2017). Multi-bound tree search for logic-geometric programming in cooperative manipulation domains. In *2017 IEEE International Conference on Robotics and Automation, ICRA 2017, Singapore, Singapore, May 29 - June 3, 2017*, pages 4044–4051. IEEE.