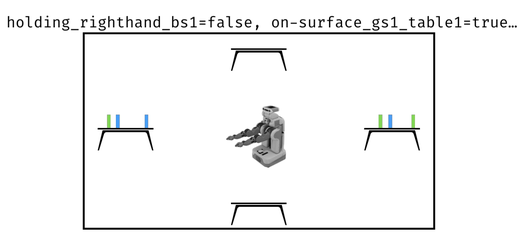
A Flexible Sampling-Based Approach to Integrated Task and Motion Planning

Wil Thomason and Ross A. Knepper
Department of Computer Science, Cornell University

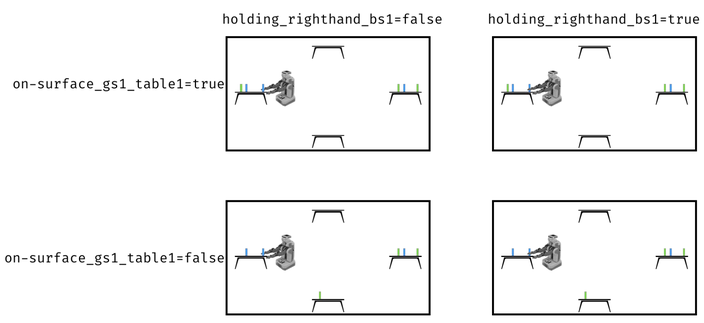
Approach

Plan in composite space

$$\mathscr{C} = C_R \underset{O}{\times} SE(3) \underset{B}{\times} \mathbb{Z}_2$$

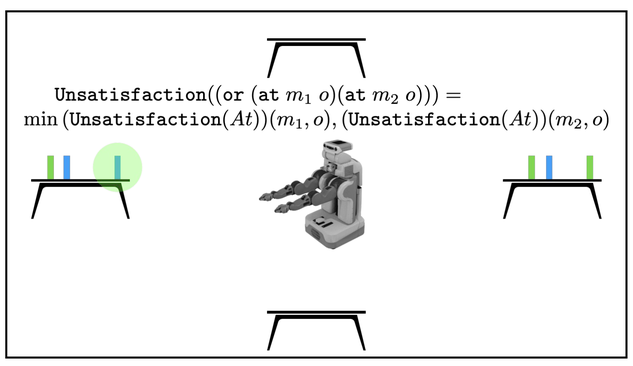


- We plan in a **composite space** containing all symbolic and continuous state for a problem.
- This makes solving TAMP equivalent to motion planning in composite space.



- We factor composite space by unique settings of symbolic state variables. Each unique setting is its own "**universe**" - a copy of the continuous state space.
- This factorization makes the TAMP problem equivalent to finding a collision-free motion plan between the initial universe (and state) to the goal universe (and optionally state).

Unsatisfaction semantics guides to precondition regions



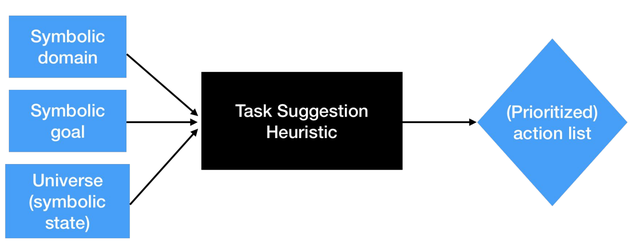
- We need to find regions of continuous space where we can take **symbolic actions**. This requires two insights:
  1. We can partition task-domain predicates into **geometric** and **symbolic** predicates - predicates which do or do not (respectively) have a reasonable interpretation in terms of a test on continuous states.
  2. We introduce a real-valued semantics for geometric predicates expressing how far a given state is from the nearest state which satisfies that predicate. We refer to this semantics as **unsatisfaction semantics**:

$$\begin{aligned} a =_\epsilon b &:= |a-b| \leq \epsilon & a =_\epsilon b &\implies \epsilon = |a-b| \\ a <_\epsilon b &:= a < b + \epsilon & a <_\epsilon b &\implies \epsilon = a - b \\ a \leq_\epsilon b &:= a \leq b + \epsilon & a \leq_\epsilon b &\implies \epsilon = a - b \\ a >_\epsilon b &:= a > b - \epsilon & a >_\epsilon b &\implies \epsilon = b - a \\ a \geq_\epsilon b &:= a \geq b - \epsilon & a \geq_\epsilon b &\implies \epsilon = b - a \end{aligned}$$

$$p \wedge_\epsilon q := \sqrt{\max(p,0)^2 + \max(q,0)^2}$$
$$p \vee_\epsilon q := \min(p, q)$$

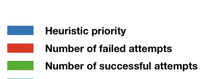
Unsatisfaction semantics provides a **navigation function** to regions of continuous space where arbitrary predicate formulae hold.

Choose actions with a black-box heuristic

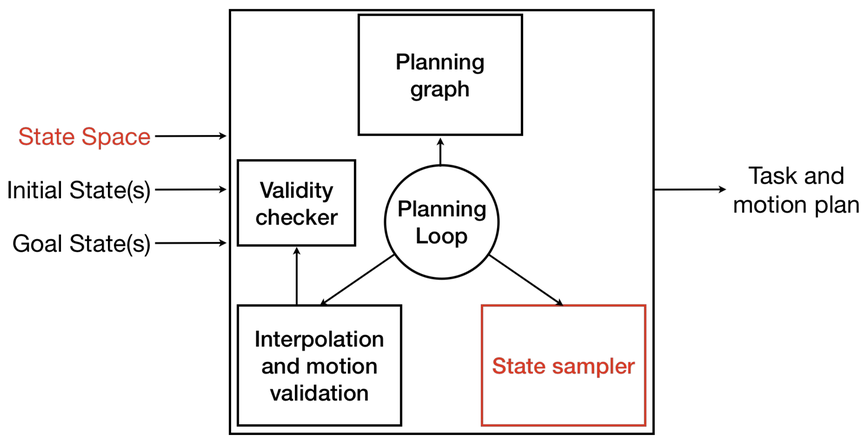


To decide what actions to attempt, we use a black-box heuristic implementing a minimal interface. We incorporate geometric feasibility information by adaptively reprioritizing suggested actions:

$$P(a) \propto \frac{\texttt{Priority}(a)}{(1 + \texttt{Fail}(a))(1 + \gamma * \texttt{Success}(a)) - \gamma * \texttt{Fail}(a)\texttt{Success}(a)}$$

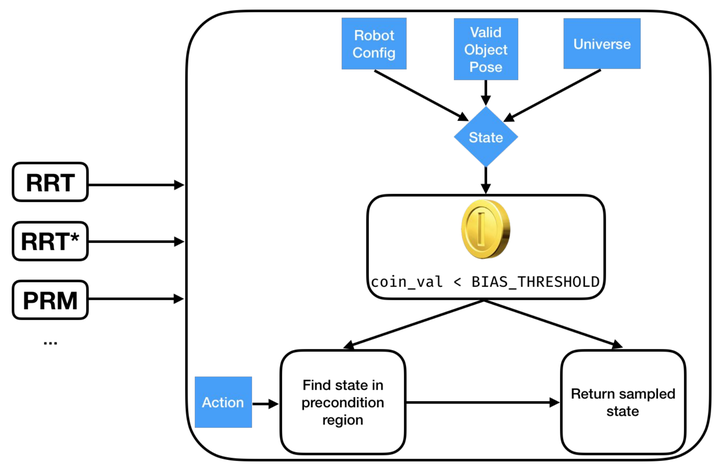


In Brief



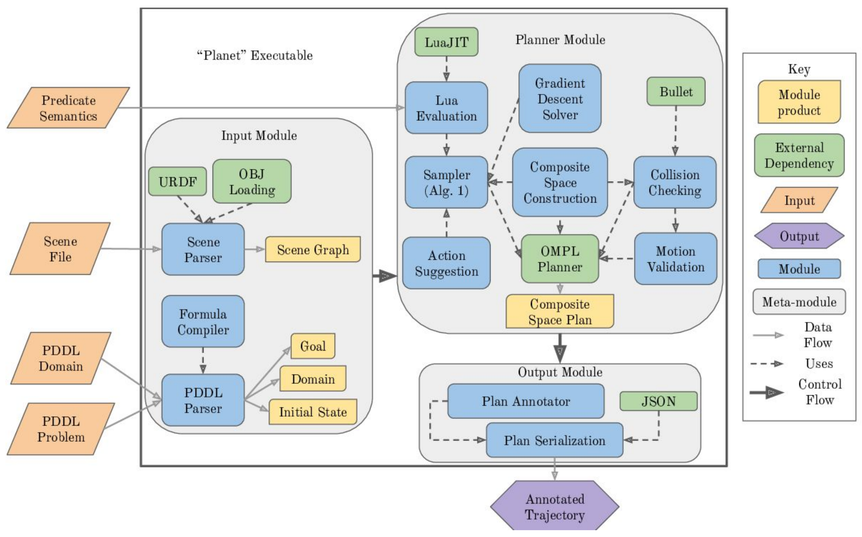
We propose a new approach to integrated task and motion planning based on the idea that **TAMP is equivalent to motion planning in an appropriate state space**. We construct such a space and demonstrate how we can solve TAMP problems by swapping out the state sampler and state space (and associated machinery, e.g. distance computation) in an ordinary sampling-based motion planner

Sampling-based architecture



- We combine the components of our approach into a **composite state sampler**. This biased-coin sampler chooses between ordinary uniform state sampling and **heuristic-guided sampling**: sampling a state in the precondition region for an action suggested by the action suggestion heuristic.
- We only modify the state sampler, so our approach can be combined with any sampling-based motion planning algorithm to adapt it into a task and motion planner.

Implementation



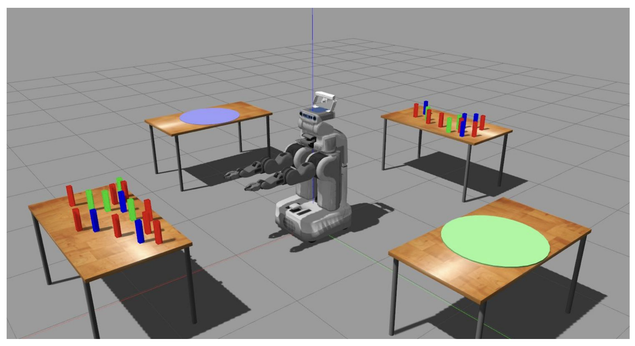
We have implemented our approach in a proof-of-concept system for evaluation. The implementation, which is written in C++, chooses ordinary RRT [**6**] as its sampling-based motion planner and the FF [**4**] "helpful actions" heuristic as its action suggestion engine. We use motion primitives from OMPL [**3**], Bullet [**1**] for collision checking, and specify problems in the format proposed by Lagriffoul et al [**2**].

We require geometric predicates to provide simple Lua implementations, which we use for state testing and reverse-mode automatic differentiation to sample in precondition regions for actions.

Future Directions

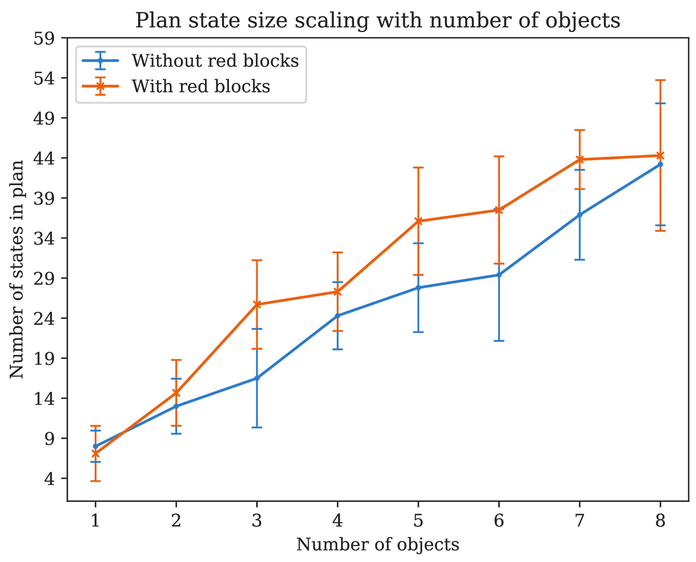
- One of the most interesting features of our approach is its agnosticism to the planning algorithm, heuristic, and predicate semantics used. As such, we plan to experiment with various combinations of these to see how performance is affected.
- Our approach also lends itself to extensions for robust execution, via predicate semantics implementations for detecting unexpected symbolic states and easy replanning by connecting the new state to the existing planning structure.
- In this work, we tried to keep our assumptions weak and lightweight. We plan to investigate the performance vs. generality tradeoff of making stronger assumptions about the problems we will solve.
- Finally, we plan to increase the generality of unsatisfaction semantics to allow a broader class of geometric predicates, as well as exploring symbolic subspace sampling for improved robustness, performance, and generality.

Evaluation

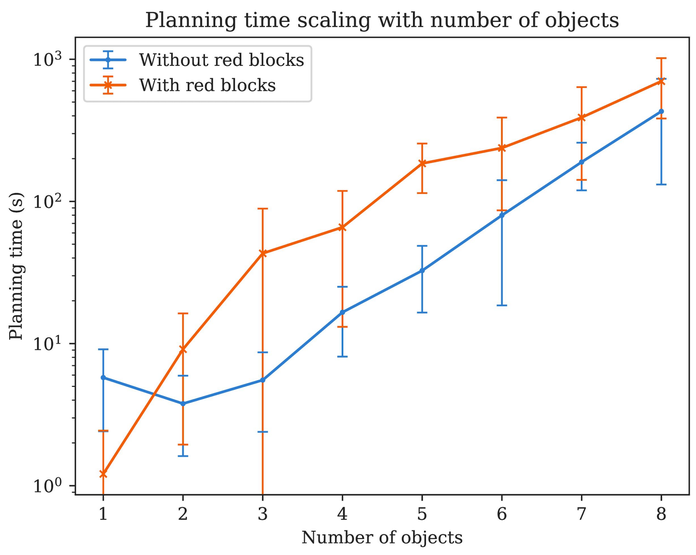


We evaluate on problems from Lagriffoul et al. [**2**], primarily the "Clutter" problem. In this problem, shown above, a PR2 must move all of the blue and green sticks to the blue and green circles (respectively) while dealing with the red sticks obstructing grasps of the others.

We evaluate both on the original problem and a variant of the problem with the red sticks removed. We include this second variant to more directly illustrate our technique's scaling with the number of goal-relevant objects.



Our planner does not produce optimal solutions (our optimality is dependent on the optimality of the motion planning algorithm chosen), but the size of the plans we produce scales linearly with the number of objects to be manipulated, as expected.



The planning time for our implementation is competitive with state-of-the-art TAMP solvers. These results have a few interesting properties:

- Planning time scales exponentially. This is expected, and largely due to the combinatorial nature of the problem. However, part of this growth likely stems from RRT's bias toward exploration, which may cause us to visit unnecessary symbolic states.
- Including red blocks does impact planning time but does not significantly impact the scaling of planning time.
- This implementation uses naive choices for planner (RRT [**6**]) and heuristic (FF's helpful actions [**4**]). We believe that we can significantly improve performance by choosing a more sophisticated planner (e.g. KPIECE [**7**], RRT-Connect [**5**]) and/or heuristic.
- The variability in planning times for our system is large. This comes directly from our choice of planner; RRT is known to have large variance in planning time and we do not use any mitigations (e.g. random restarts) for this issue.

References

[**1**] Erwin Coumans et al. Bullet Physics Library. 2013.
[**2**] Fabien Lagriffoul, Neil T. Dantam, Caelan Garrett, Aliakbar Akbari, Siddharth Srivastava, and Lydia E. Kavraki. "Platform-Independent Benchmarks for Task and Motion Planning". In: IEEE Robotics and Automation Letters 3.4 (Oct. 2018), pp. 3765–3772. ISSN: 2377-3766, 2377-3774.
[**3**] Ioan A. ¸Sucan, Mark Moll, and Lydia E. Kavraki. "The Open Motion Planning Library". In: IEEE Robotics & Automation Magazine 19.4 (Dec. 2012), pp. 72–82. DOI: 10.1109/MRA.2012.2205651.
[**4**] Jörg Hoffmann and Bernhard Nebel. "The FF Planning System: Fast Plan Generation through Heuristic Search". In: Journal of Artificial Intelligence Research 14 (2001), pp. 263–312. ISSN: 10769757.
[**5**] Kuffner Jr, J. J., & LaValle, S. M. (2000, April). RRT-connect: An efficient approach to single-query path planning. In ICRA (Vol. 2).
[**6**] Steven M. Lavalle. "Rapidly-Exploring Random Trees: A New Tool for Path Planning". (May 1999).
[**7**] Şucan, I. A., & Kavraki, L. E. (2009). Kinodynamic motion planning by interior-exterior cell exploration. In Algorithmic Foundation of Robotics VIII (pp. 449-464). Springer, Berlin, Heidelberg.