Mechanical & Industrial Engineering
UNIVERSITY OF TORONTO

Scalable Planning with Tensorflow for Hybrid Nonlinear Domains

Ga Wu, Buser Say, Scott Sanner

Key Questions

How can we plan effectively (nb, planning not learning!) in hybrid planning problems with (piecewise) nonlinear dynamics with 576,000 actions distributed over a horizon of 95 time steps and 100 parallel instances in 4 minutes?

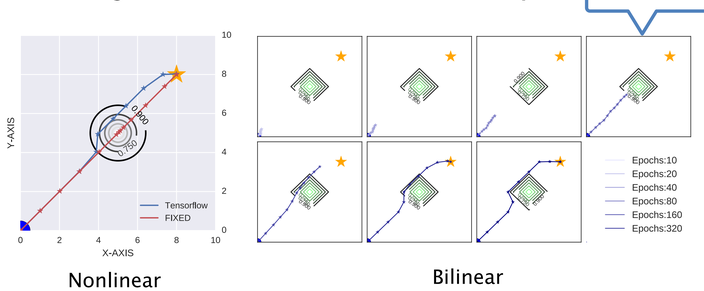
Can we exploit efficient compilation of symbolic specifications (e.g., RDDL), auto-differentiation, GPUs and recent advances in non-convex gradient descent such as RMSProp and Adam available in Tensorflow?

How does our Tensorflow-based planner compare to the optimal solution (when it can be computed) and state-of-the-art hybrid nonlinear planners?

Result Highlights

2D Navigation in continuous domain and action space

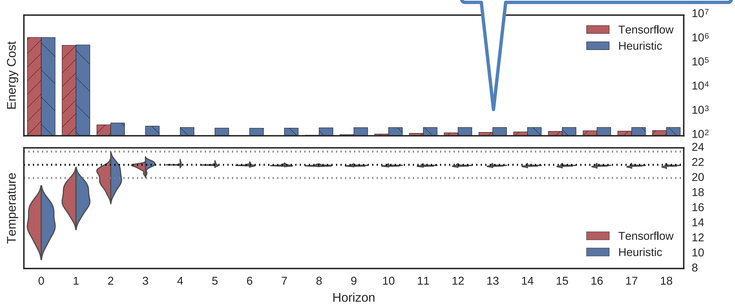
Path Planning with Gradient Descent



Nonlinear Bilinear

HVAC control that optimizes energy usage

Optimal Planning without Sacrifice Comfort



Energy vs Temperature

Domain Definition Language to Tensorflow Graph

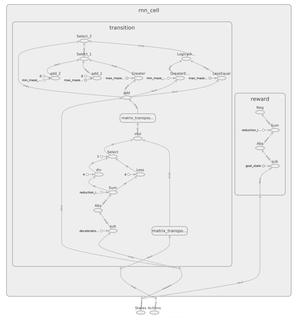
Linear time / space translation from symbolic hybrid planning languages like RDDL to Tensorflow symbolic objects:

RDDL(Sanner, 2010):

```
cpfs {
      distance = [sum_{?l:dim}[abs[location(?l)-CENTER(?l)]]];
      scalefactor(?l) = if(distance<=2.0) then distance/2.0
                        else 1.0;
      proposedLoc(?l) = location(?l) + move(?l)*scalefactor(?l);
      location'(?l)= if(proposedLoc(?l)<=MAXMAZEBOUND(?l) ^ proposedLoc(?l)>=MINMAZEBOUND(?l)) then proposedLoc(?l)
                     else (if(proposedLoc(?l)>MAXMAZEBOUND(?l)) then MAXMAZEBOUND(?l) else MINMAZEBOUND(?l));


};

reward = - sum_{?l: dim}[abs[GOAL(?l) - location(?l)]];

state-action-constraints {
      forall_{?l:dim} move(?l)<=MAXACTIONBOUND(?l);
      forall_{?l:dim} move(?l)>=MINACTIONBOUND(?l);
      forall_{?l:dim} location(?l)<=MAXMAZEBOUND(?l);
      forall_{?l:dim} location(?l)>=MINMAZEBOUND(?l);
};
```

Directly compile constraints into graph

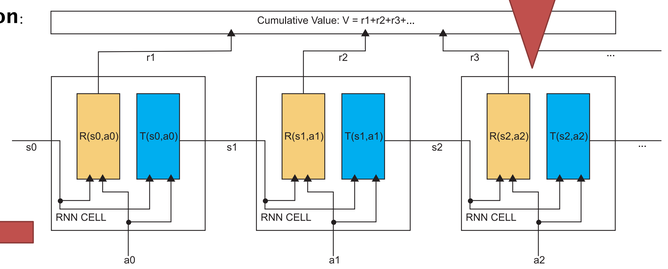
Tensorflow Graph:



RNN Like Model Structure

Optimize actions via backpropagation:

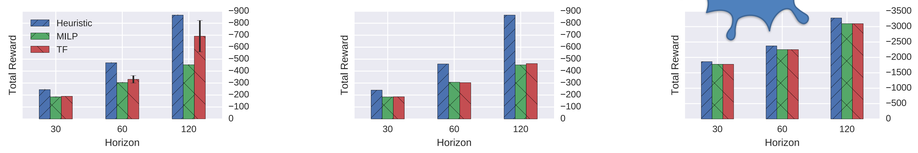
$$\frac{\partial L}{\partial a_{itj}} = \frac{\partial L}{\partial L_i}\frac{\partial L_i}{\partial a_{itj}} = \frac{\partial L}{\partial L_i}\frac{\partial \mathbf{s}_{it+1}}{\partial a_{itj}} = \frac{\partial L}{\partial L_i}\frac{\partial \mathbf{s}_{it+1}}{\partial a_{itj}}\sum_{\tau=t+2}^{T}[\frac{\partial L_i}{\partial r_{i\tau}}\frac{\partial r_{i\tau}}{\partial \mathbf{s}_{i\tau}}\prod_{\kappa=\tau}^{t+2}\frac{\partial \mathbf{s}_{i\kappa}}{\mathbf{s}_{i\kappa-1}}].$$



Comparing to MILP and Fmincon Solver

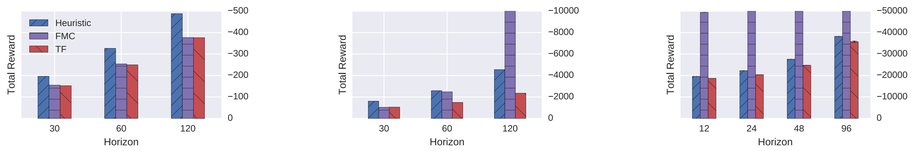
lower is better

Comparing with optimal MILP in linear domains



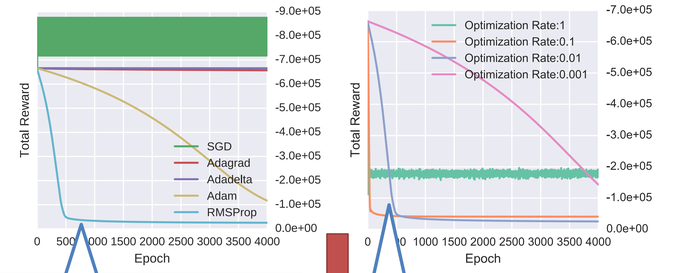
Navigation-Linear Navigation-Bilinear Reservoir-Linear

Comparing with interior point Fmincon solver in nonlinear domains



Navigation Reservoir HVAC

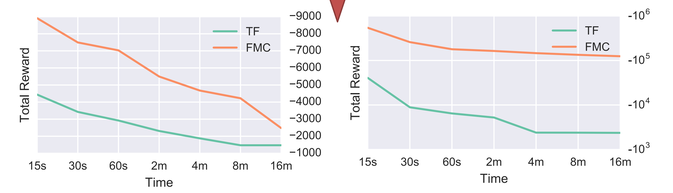
Working with Advanced SGD Optimizers



RMSProp substantially outperforms other optimizers

Optimization rate = 0.01 works well across multiple domains

Faster Convergence



Comparing to Hybrid Planners

Unfortunately, no existing solver can solve any of the domains we tested, including ENHSP(Scala, 2016)

Conclusion

We introduced a new Tensorflow-based planner for hybrid nonlinear domains that exploits symbolic domain structure, often outperforms state-of-the-art planning approaches, and can scale to extremely large domains leveraging GPUs.

The use of RMSProp seems critical for fast convergence to a good solution in these nonlinear domains.

We conjecture that RMSProp with auto-differentiation directly on the piecewise nonlinear dynamics better exploits domain structure than existing constrained optimization compilations.